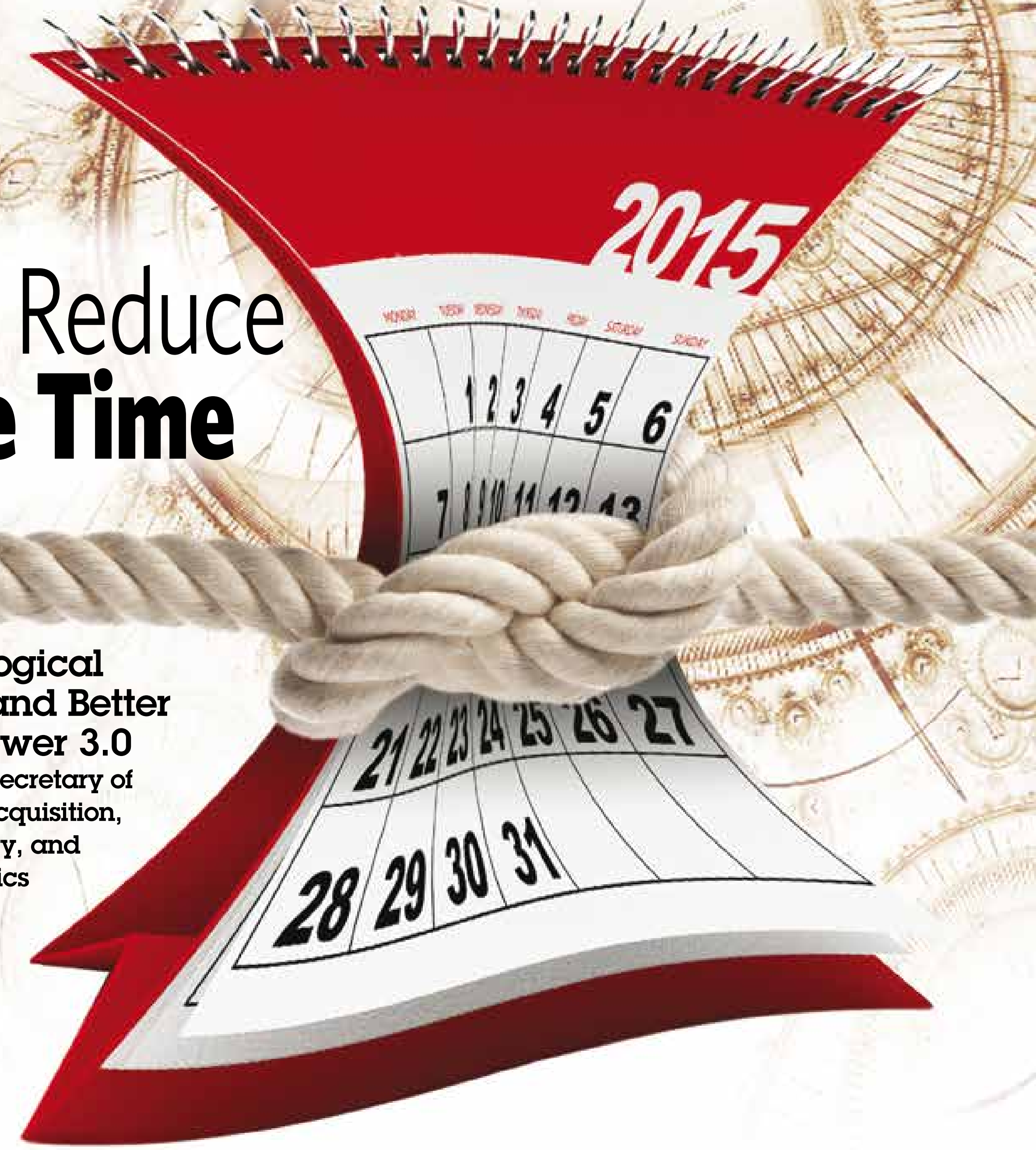
www.dau.mil | November-December 2014

Defense AT&L

Acquisition, Technology and Logistics

A PUBLICATION OF THE DEFENSE ACQUISITION UNIVERSITY



Please Reduce **Cycle Time**

Technological Superiority and Better Buying Power 3.0
by the Under Secretary of Defense for Acquisition, Technology, and Logistics

The Big IDEA
Dynamic Stakeholder Management

Think Portfolios, Not Programs

Mining Hidden Gems
Extract Information Systems' Value

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **DEC 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**
4. TITLE AND SUBTITLE **Defense AT&L Magazine. Volume 43, Number 6. November-December 2014**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Acquisition University,9820 Belvoir Road,Fort Belvoir,VA,22060-5565**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**52**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

CONTENTS

ALSO

Defense
AT&L

Vol XLIII No. 6, DAU 241

Published by the
DEFENSE ACQUISITION UNIVERSITY

Article preparation/submission guidelines are located on the inside back cover of each issue or may be downloaded from our website at <http://www.dau.mil/pubscats/pages/defenseatl.aspx>. Inquiries concerning proposed articles can be made by e-mail to datl@dau.mil or by phone to 703-805-4282 or DSN 655-4282.

Subscribe/unsubscribe/change of address: Fill out, sign, and mail the postage-free self-mailer form at the back of this issue, or download the form at <http://www.dau.mil/pubscats/pages/defenseatl.aspx>.

Privacy Act/Freedom of Information Act
If you provide us your business address, you will become part of mailing lists that are public information and may be provided to other agencies upon request. If you prefer not to be part of these lists, use your home address. Do not include your rank, grade, service, or other personal identifiers.

Defense AT&L (ISSN 1547-5476), formerly *Program Manager*, is published bimonthly by the DAU Press and is free to all U.S. and foreign national subscribers. Periodical postage is paid at the U.S. Postal Facility, Fort Belvoir, Va., and additional U.S. postal facilities.

POSTMASTER, send address changes to:
DEFENSE AT&L
DEFENSE ACQUISITION UNIVERSITY
ATTN DAU PRESS STE 3
9820 BELVOIR ROAD
FT BELVOIR VA 22060-5565

Technological Superiority and Better Buying Power 3.0

Frank Kendall

Each morning I start my day with a half hour or more devoted to reading the latest intelligence. I've been doing this for about four-and-a-half years now. It took me only a few weeks from the time I came back into government in March 2010 to realize that we had a serious problem. Some of the countries that might be future adversaries, (or that could at least be counted on to sell their weapons to countries that are our adversaries) were clearly developing sophisticated weapons designed to defeat the United States' power-projection forces. Even if war with the United States were unlikely or unintended, it was quite obvious to me that the foreign investments I saw in military modernization had the objective of enabling the countries concerned to deter regional intervention by the American military.

How did we get here? This journey began after the Cold War and in particular the First Gulf War that followed shortly thereafter. At that time, I was the Director of Tactical Warfare Programs in the Office of the Under Secretary of Defense for Acquisition. For years, since the 1970s, the Department had been working on a suite of capabilities originally designed to overcome the Soviet numerical advantage in Europe. As a young Army officer, I had served in Germany in the 1970s and studied firsthand the problem

that successive echelons of Soviet armor formations posed to NATO forces. Our answer to this problem was something called Follow-On-Forces-Attack (FOFA), which had grown out of the Assault Breaker technology demonstration program at the Defense Advanced Research Projects Agency. The basic idea was to combine wide-area surveillance, networked Command, Control and Communications, and precision munitions into an operational concept that would negate the Soviet numerical advantage. The concept could be summed up as "one shot, one kill." From 1989 to 1994, I was responsible for the FOFA programs. In the First Gulf War, we had a chance to demonstrate the effectiveness of this concept, and we did so.

As we started operations against Saddam Hussein, most experts predicted thousands of coalition casualties. In the event, the number was only a few hundred. The combination of sensors like the JSTARS [Joint Surveillance Target Attack Radar System] and precision munitions like Maverick and laser-guided bombs made quick work of Iraqi armor formations. Stealth also was introduced to the battlefield to great effect by the F-117.

The dramatic success of American and coalition forces in 1991 did not go unnoticed. No country paid more attention to this stunning display of military dominance than China, followed closely by Russia. The First Gulf War marked the beginning of a period of American military dominance that has lasted more than 20 years. We used the same capabilities, with some notable enhancements, in Serbia, Afghanistan, Libya and Iraq. It has been a good run, but I am concerned that, unless we act quickly, this period will end in the not-too-distant future.

When I left the Pentagon in 1994, the intelligence estimates suggested that, while China might be a concern in the future, the United States then had no reason to be worried for 15 to 20 years. It is now 2014, and I am worried. There has been more than adequate time for countries like Russia, with its energy-revenue-funded military modernization, and China, with its spectacular economic growth, to develop counters to what has been called either the Military-Technical Revolution or the Revolution in Military Affairs that the United States introduced so dramatically in 1991.

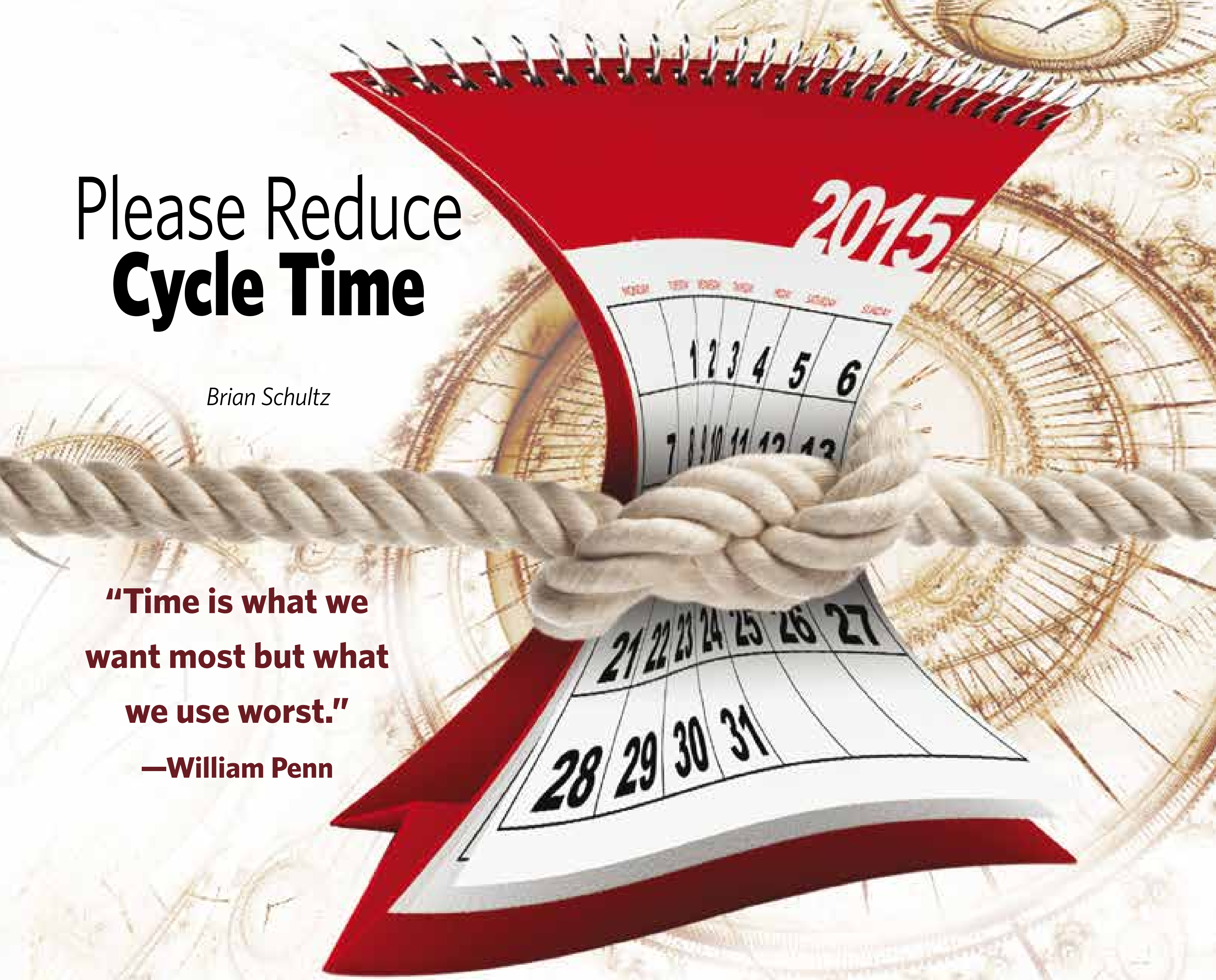
The foreign modernization programs that I refer to include investments in cyber capabilities, counter-space systems, electronic warfare programs, land-and-surface-ship attack ballistic and cruise missiles with smart seekers, anti-air weapons, advanced platforms to host these capabilities and many more. Taken together, these modernization programs are clearly designed to counter American power projection forces and to ensure that the United States does not interfere in the areas close to Russia or China. Even if our relationships with these states improve and military confrontation is avoided, the capabilities I am concerned about will still quickly proliferate to other states, such as Iran and North Korea. We cannot afford to be complacent about our technological superiority, and we cannot allow other less-sophisticated threats to distract us from the task of maintaining that superiority. This brings us to Better Buying Power 3.0.

We cannot afford to be complacent about our technological superiority, and we cannot allow other less-sophisticated threats to distract us from the task of maintaining that superiority.

For the last four years, our focus in Acquisition, Technology, and Logistics has been on improving our business outcomes. Usually, we discuss the Better Buying Power goals in terms of productivity, waste elimination, better business deals, and efficient execution of programs and services. In BBP 3.0, my goal is to shift our emphasis toward the actual products we are developing, producing, fielding and maintaining. We will continue our efforts to improve productivity, but the focus of BBP 3.0 is on the results we are achieving—particularly our ability to bring innovative and game-changing technologies into fielded capabilities for the warfighter as quickly and efficiently as possible. Our technological superiority is not assured. I also do not expect the budget climate to improve for the foreseeable future. Sequestration may well return in Fiscal Year 2016—and, even if it does not, the threat is unlikely to be removed entirely.

We are going to have to work hard to bring the innovation and technology we need to our warfighters—and we are going to have to achieve this in a very tough environment.

Please Reduce **Cycle Time**



Brian Schultz

"Time is what we want most but what we use worst."
—William Penn

As William Penn noted centuries ago, time might be our most precious resource but it is also one that we have trouble managing effectively. While cost-performance trade-offs get a lot of emphasis in developmental acquisition efforts, schedule or cycle time is also an important part of the cost-schedule-performance triad that determines outcomes. Note that the terms "cycle time" and "schedule" will be used interchangeably in this article to mean the total time required from program initiation until Initial Operational Capability (IOC).

Acquisition cycle time continues to be a "hot topic." Over years, many have argued that it simply takes too long to get capability to the warfighter and that fundamental reform is needed to address this issue. More recently, we see the imperative to deploy capabilities faster in light of cyber and asymmetric threats. Several studies have validated this notion that it is taking longer now than in past decades to develop and field Department of Defense (DoD) weapon systems. Despite all the attention and reforms, the issue has not gone away. In fact, it may even be more problematic now than in the past because of program complexity, use of new and advanced materials, software-intensive designs, advanced manufacturing techniques and many other factors.

Schultz *is a professor of program management at the Defense Acquisition University's Mid-Atlantic Region in California, Md.*

A DoD Inspector General audit report (No. D-2002-032, Dec. 28, 2001) identified that in 1960 major defense acquisition programs (MDAPs) required seven years for completion, again defined as program start to IOC. In 1996, this metric had grown to 11 years. A more recent Government Accountability Office study (GAO-14-145T) highlighted that the average delay in achieving IOC on MDAPs had grown from 22 months in 2008 to 27 months in 2012 while cost growth increased from $323 billion to $411 billion. Although the purpose of this discussion is not to examine the history or causes of acquisition cycle time growth, it is important to understand why there is such an emphasis on schedule and getting capability to the warfighter more rapidly.

Cycle time can be addressed in the context of "micro" and "macro" perspectives. Micro-cycle time is defined here as the specific program schedule tasks and dependencies necessary to get a capability fielded, based on the program's unique technical and programmatic aspects. Thus, it is addressed in the context of a specific program and is the responsibility of the program manager (PM) to plan and execute—including any programmatic assumptions, constraints and logic.

Implementing concurrent efforts and/or eliminating some tasks are often used to enable this rapid approach, recognizing that risk and program inefficiencies may increase.

On the other hand, macro-cycle time is defined as the impact that the overall acquisition management and decision-support-system structures have on the time it takes to field a capability from Milestone B to IOC. For example, macro-cycle time considerations would include time necessary for the processes and approvals within the DoD decision-support systems.

Macro-cycle time considerations are addressed in statutory, regulatory and policy documents such as the new interim DoDI 5000.02. Note that the new 5000.02 Instruction includes an Accelerated Acquisition Program model, where schedule considerations override other programmatic constraints. Implementing concurrent efforts and/or eliminating some tasks are often used to enable this rapid approach, recognizing that risk and program inefficiencies may increase.

Another aspect of macro-cycle time considerations in the 5000.02 is tailoring each program based on the unique aspects associated with it. The Accelerated Acquisition Program model is called out as one specific example of tailoring, and many others are possible. The basic premise suggests that PMs should look for opportunities to structure their programs in a manner that reduces time and cost, while accepting reasonable risks. The "will" and "shall" statements in the regulatory guidance do not necessarily override the mandate for a tailored approach. So, even if one is not using an accelerated model, opportunities to accelerate the program schedule should be explored.

Both macro- and micro-cycle time aspects should be addressed as part of the overall risk- and opportunity-management framework. The following discussion provides some examples of cycle-time risks and opportunities based on my experiences. A robust and continuous approach to assessing and managing schedule risks and opportunities can be very useful in getting to a successful acquisition outcome and help answer the question, "What can we do to reduce cycle time?"

Risks

Schedule logic and assumptions: PMs typically build a Government Roadmap Schedule and an Integrated Master Plan (IMP) that outline the overall schedule for the program and key events and criteria to complete the events. The IMP and Government Roadmap Schedule provided to the contractor may include hard dates that contractors must meet. These program constraints are then used as the basis for the contractor-developed Integrated Master Schedule (IMS) that establishes dates and schedule-task relationships for the program execution.

In a competitive environment, companies may be reluctant to identify an overly aggressive schedule as high risk since this could jeopardize their competitive positions. PMs should encourage open dialogue with industry in the pre-award planning stage to assess the amount of time needed to complete the planned work and to validate schedule assumptions before contract award. If we decide to take on a high-risk schedule, at least we can manage it as such and ensure actions are planned to address contingencies if the risks are realized.

The logic associated with the schedule relationships should also be carefully reviewed and periodically revisited. This logic can be flawed and change over time as we learn more and better understand the schedule interrelationships. It may also be wise to keep the complexity at a manageable level.

I learned this lesson while managing a software-intensive program. We decided to create a very detailed master schedule with multiple supporting subschedules that linked and

automatically updated the master schedule if all the inputs were entered correctly. The effort was well intentioned as the program involved a large team of developers and engineers working concurrently on several modules. However, the schedule and subschedules became so complicated and difficult to manage that they became unusable. We ended up reverting to a simpler schedule that provided enough oversight to keep on top of key tasks and actual progress. We also adjusted the schedule several times based on the knowledge of developer velocity and features that could be descoped and/or deferred.

Use of Commercial Off-the-Shelf (COTS) products: Using COTS offers many benefits and is prescribed in DoD Directive 5000.01 as the preferred approach to satisfying user needs. COTS can also present risks to a program and has been at the heart of significant cost and schedule delays within DoD. The issue has not been the COTS product itself but rather attempts to modify it and/or not fully understanding the COTS product.

A Dec. 23, 2013, Reuters article, "Why the Pentagon's accounting fixes end up broken," highlighted a common thread of several failed Enterprise Resource Planning (ERP) projects. A COTS product is chosen for an ERP solution but is then modified to reflect the legacy-system business model. These COTS modifications create a unique software product that is no longer COTS and becomes difficult to maintain. Furthermore, the benefits of COTS—such as the ability to leverage upgrades, training and support—can be lost.

Years ago, I observed an ERP system implementation that encountered this exact model. The modified COTS software worked and passed the acceptance tests but never was implemented by the customer due to the issues associated with maintaining it. Other programs apparently have learned this same lesson and the word is out that, if you decide to use COTS, you need to adopt the business process model it is based on rather than try to keep your existing processes in place as part of the COTS implementation.

For hardware, COTS can also present some risks. Many programs use COTS computers and servers, even as part of their mission-computing design. Since these items can quickly become obsolete and no longer supported by the original manufacturer, PMs should plan appropriate mitigations, including the use of periodic technology updates.

Inadequate planning: The imperative to accelerate can backfire and actually be counterproductive if not planned and managed properly. A good example is the use of Undefinitized Contract Actions (UCAs) to accelerate the start of development. On the surface, one might expect a faster overall schedule since it avoids the often lengthy upfront process of proposal development and negotiations before work starts. However, according to the Under Secretary of Defense for Acquisition, Technology, and Logistics' 2013 *Annual Report on the Performance of the Defense Acquisition System*, UCAs are identified as contributors to cost and schedule growth in DoD development contracts. UCAs were not correlated with cost and schedule growth in early production.

Another example is rushing the contracting cycle to stay on schedule. This often starts with the release to industry of the request for proposal (RFP). Given the lead time involved in contracting cycles, the temptation can arise to accelerate the RFP development and release, bypassing internal reviews and/or skipping a draft RFP release for industry comment. Some might refer to this behavior as schedule driven versus event driven, as detailed in Dr. Mark Husband's March-April 2014 *Defense AT&L* article, "Schedule or Event Driven?"

Every time I have tried to accelerate an RFP release, it ended up costing more time to correct issues such as inconsistencies or lack of clarity in RFP requirements. Industry will provide valuable feedback in a draft RFP that will often help the government team develop a better final RFP that can avoid future perturbations. While I don't have any data to back it up, my experience suggests that a very robust acquisition strategy and RFP planning effort saves time in the overall schedule and helps DoD get better outcomes.

Opportunities

Concurrency: I had a great experience working an aircraft mission-system upgrade program that employed a concurrent development and production strategy to accelerate the cycle time for fielding. While this strategy was risky, the risks were identified and managed jointly by the contractor and government team in a robust and transparent manner. When the program was planned, we expected that the software design would require several builds and iterations after a series of both ground and flight test events.

Meanwhile, the hardware designs were expected to be stable since we were integrating proven avionics, sensors and communication subsystems. A concurrent strategy was adopted, but only after careful analysis of the program plan, schedule and technical risks.

The teamwork and commitment of the joint DoD-contractor team played a big role in the success. Despite some bumps along the way, we delivered the capability within budget and on schedule. Note that due to the risks of this approach, a concurrent strategy is likely to get significant scrutiny and should be used only where all the right conditions are in place. These conditions include the right expertise, adequate resources (both human and financial), risks that are assessed as no higher than moderate, and buy-in from the entire team, including top management of the DoD and contractor teams.

Schedule is an important message: It may seem an oversimplification but sending a message to appropriate stakeholders, including the contractor, that schedule is important should not be overlooked. While cost-performance trades continue to get a lot of emphasis, how about addressing

schedule-performance and cost-schedule trades similar to the Agile methodology in developing software? Note that one of the tenets of Agile software development methodology is that features will be managed as a variable in a given build but schedule and cost will remain fixed. This tenet is based on the idea that missing a delivery date will create greater overall dissatisfaction and impact than deferring some functionality until a later build.

have learned and what we can improve. Cycle-time reduction will be difficult to achieve without an organizational culture of identifying and managing those opportunities.

Challenging the status quo and creating an environment where performance is rewarded can enable schedule compression opportunities. A few years ago, I worked on an urgent program to get a radar system installed in Iraq. When we looked at the

Challenging the status quo and creating an environment where performance is rewarded can enable schedule compression opportunities.

normal production lead time to get the radar produced and fielded, it was impossible to meet the need date. The project manager suggested that we tell the user we could not meet its requirement.

We then brainstormed and asked if the radar was currently in production and if we could divert another user's delivery with payback from the system we ordered. Sure enough, one of the users was happy to divert its planned delivery, enabling us to meet the compressed timeline. We also had to accelerate and combine some site-activation efforts and enlist support from other agencies to help us obtain access and eventual safety and accuracy certifications.

On another program, I observed an industry PM question why we needed so much documentation as part of a draft RFP. The documentation in question related to COTS items where a lengthy review of specifications added no value. This simple suggestion saved a lot of time and effort on work that the previous teams always did.

Another practice involves setting clear expectations at the beginning of a new contract. Assuming a fixed-price type contract, it should be clear that missing a contract deliverable date is a serious breach. What some may not realize is that if the government allows schedule slips without consequences, the message is clear—that schedule is not important.

I once was told by a senior contracting official that accepting several late deliveries and then deciding to take action after the fact is too little, too late. It is the equivalent of saying that schedule is not important. Rather, the message should be clear that, if the contractor expects to miss any contractual delivery date, notice is to be given before the slip occurs accompanied with an explanation and get-well plan. This enables the two parties to discuss and attempt to resolve the issue before the slip and lets the contractor know that we expect performance in accordance with the contract. Appropriate corrective actions and contractual remedies also can be considered. Finally, the contractor's performance will be documented in the Contractors' Performance Assessment Report, which can be an important factor in future source selections.

Program teams have several tools to help manage the schedule and assess opportunities to accelerate. Some companies are using the theory of constraints (originated by Eli Goldratt, author of the book *The Goal*) as a basis for lean project management and lean manufacturing techniques to drive accelerated schedules and cost reductions. PMs need to understand what methodology their industry counterparts are using and why they believe it's appropriate.

Challenge the status quo: There is an old adage that goes something like "the schedule will expand to fit what we planned (even if we learn we can do it faster)." This humorous saying highlights an important opportunity when assessing a program schedule and looking for ways to get it done faster. The opportunity is to take a look at what we are doing, what we

Conclusion: Cycle time is one of the key pillars of acquisition and has direct links and impacts to other programmatic elements. PMs must navigate through both macro and micro aspects of cycle-time risk and opportunities. The imperative to field systems and solutions quicker is challenging and will probably become more so, given the threat environment and pace of new technology. PMs should create an environment and expectation that cycle time is important in all aspects of acquisition processes and tasks, while ensuring credible execution to the baseline schedule!

All this cycle-time talk reminds me of the user who said to me: "We needed this capability yesterday. Why does it take so long?"

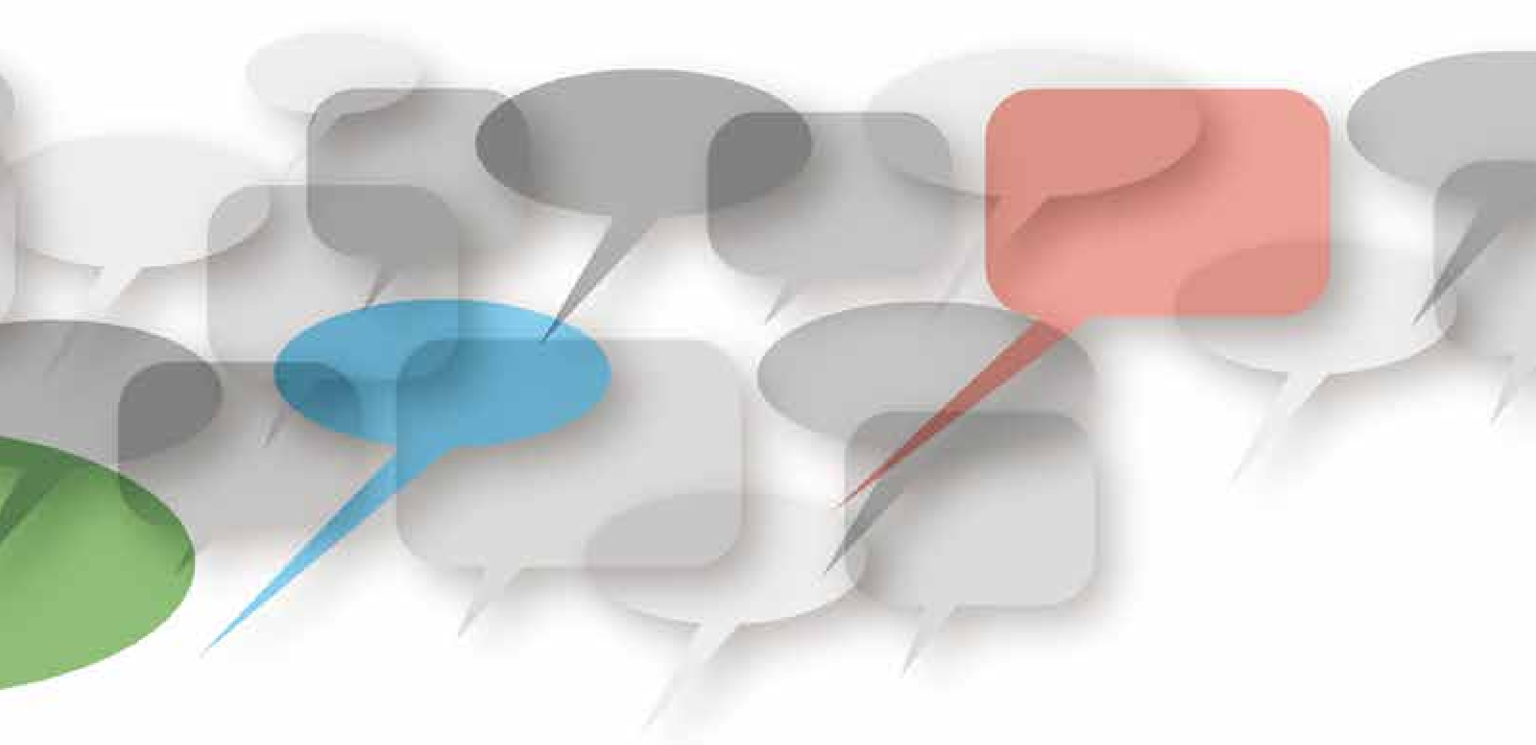
The author can be contacted at **brian.schultz@dau.mil**.

The Big IDEA

Dynamic Stakeholder Management

Lt. Col. Franklin D. Gaillard II, USAF
Frank Gaillard, Ph.D.

Lt. Col. Gaillard, *a graduate of the U.S. Air Force Academy, is the materiel leader and program manager of the E-3 AWACS Block 40/45 program at Hanscom Air Force Base, Mass. His father and coauthor,* **Dr. Gaillard**, *is a retired U.S. Navy lieutenant commander and currently a professor of information systems at Global Campus, Troy University.*



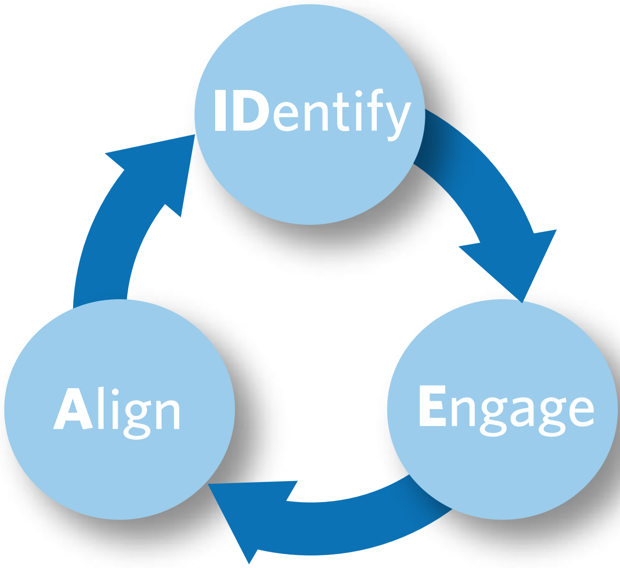
Who holds a stake in your program? What are their interests? Would your program flourish or spiral downward without their advocacy?

In today's dynamic environment, program managers (PMs) and acquisition professionals, across a variety of sectors and disciplines, are increasingly subject to a wide variety of pressures and constraints. Program managers must balance the perspectives, interests and motivations of a variety of organizations both internal and external to the program office in order to achieve program goals. There are relationships with the end user employing the system being acquired, fielded and sustained as well as interaction with the defense industrial base that helps to develop the systems for use in end products.

Corporate staffs provide the necessary oversight of program health and guidance required to ensure compliance with applicable statute, policy and law. Depending on the phase or development level of the program, various other agencies and independent organizations may have a role in ensuring the success of acquisition efforts. As a result, PMs often are pulled in multiple directions, struggling to find the appropriate balance in tending to the needs of the myriad, increasing stakeholders and required certifications, while simultaneously leading projects and managing the program's cost, schedule and performance.

Many acquisition efforts are complex, interwoven with system complexity, applicable policy, guidance and laws and are riddled with budget constraints. To achieve program goals, program managers must make choices between potentially sacrificing program content or readdressing requirements. Therefore, in today's challenging environment, it is incumbent on leaders to develop a comprehensive stakeholder management strategy that builds support, enables advocacy and provides an underpinning for managing acquisition success. The IDEA model is an approach focused on: **ID**entifying stakeholders, **E**ngaging them early and often, **A**ligning interests and goals and completing the cycle by reassessing stakeholder relevancy, then repeating the process when necessary. Using this framework,

Figure 1. IDEA for Stakeholder Management



PMs and project managers can ensure they are the forefront of managing collective interests, while keeping pace with change and with a dynamic operating environment (See Figure 1).

Even before beginning the journey to ID, Engage and Align with stakeholder interests, PMs and project managers need to have a firm grasp or established criteria defining a stakeholder. One approach would be to define a stakeholder as any organization, person or entity that has an interest in or can affect, or be affected by, your program. A word of caution here: An organization may not be directly impacted or have the ability to directly influence a program but may still be a stakeholder. Stakeholders can be very involved in the day-to-day operations of a program, or they may stand on the periphery, getting involved much less frequently. They may travel to observe the operations of your program, participate in weekly staff meetings with your team or communicate by phone, formal correspondence or even just by e-mail.

There are two important concepts in defining stakeholders. First, there is the need to define what a stakeholder is, which is largely left to the discretion of the PM. Second, and probably most important, a stakeholder is not the PM. This would seem obvious. However, one might be surprised by how many stakeholders attempt to exert authority and control over a program management team. It is critically important to understand that the PM is held accountable for leading the team to manage cost, schedule, performance and, ultimately, the success of the program. Once those two fundamental concepts are understood, a PM is ready to begin the first step using the IDEA model for stakeholder management.

Step 1: IDentifying Stakeholders

As discussed previously, not every person, organization or agency is a key stakeholder. A PM must be able to identify and sort stakeholders and address them appropriately. Take this scenario for example: A project manager is in charge of procuring and installing a revolutionary upgrade program in a fleet of rental cars. The rental car company is a key stakeholder, with an interest in upgrading the fleet of rental cars. The rental car company ships the cars to the maintenance company (another stakeholder), which then installs the driver-assist upgrade. Another stakeholder is the supplying company that manufactures the parts and components used in the upgrade. In this scenario, a PM is faced with allocating resources (time, funding and personnel) and with coordinating a schedule with the rental car company (or end user), the maintenance organization and the supplying company to achieve the program goal of delivering rental cars with the upgrade. All these organizations could be considered key or first-tier stakeholders. Examples of other stakeholders include the corporate company leadership providing funding, direction and guidance on how to run the program; the insurance company that wants to review test reports of the system's safety; or competing units within the project lead's company—all vying for a share of limited resources. Once stakeholders have been identified, a PM is ready to begin the second step—engaging with the stakeholders.

Step 2: Engage with Stakeholders Early and Often

Engaging with stakeholders is absolutely critical to the success of any program. Waiting for an issue to arise before making your acquaintance with a stakeholder may insert unnecessary challenges, hinder communication and promote a less-than-desirable working situation. It is imperative that PMs boldly seize the initiative and proactively establish a relationship well in advance of any issue or crisis.

Once a PM has determined the need to engage, the approach needs to be considered. If possible, PMs should meet with key stakeholders in person. There is no substitute for interacting with organizations directly. This represents a valuable opportunity to gain perspective and insights into the values and interests of the stakeholder, but it also provides an opportunity for both parties to discuss their visions and goals for the project and get a first glimpse into how easy it will be to align these goals for program success (Step 3).

In the case of the rental-car upgrade program, proactive stakeholder management would entail PM visits with all the key stakeholders (the car rental agency, maintenance company and supplier company). Proactive engagement affords PMs an opportunity to gain critical insights into critical focus areas, allowing them to lead or turn issues and mitigate risks before they become problems. When in-person visits are not possible, video teleconference, telecom or e-mail can also be effective methods for gaining this needed information.

The keys to effective engagement are tailoring the engagement strategy to fit the level of stakeholder, finding an appropriate time and method of communication (in person, telecom or e-mail), and ultimately establishing a relationship that is conducive to a successful program. For a PM, the engagement plan builds the foundation for aligning interest and goals of all the stakeholders to achieve program success (Step 3).

In a constrained environment, program leadership needs to evaluate if these "perceived" stakeholders indeed still add value and if the cost of focusing resources to address their issues is outweighed by benefit to the program.

Step 3: Align Interests—What Are We All Here to Do?

Once a PM has identified stakeholders and engaged and communicated with them, the next step is to align goals and interests to support the program. Organizations all have varied interests, priorities, motivators, missions, goal and visions. In leveraging the second step of engagement, PMs get a chance to assess and understand these interests. This includes considering the perspectives and the frames of reference of the stakeholders. In the case of the rental car upgrade program, the rental agency is concerned with upgrading a fleet of rental cars and staggering this down time to allow for ongoing operations of the rental car service. The maintenance company is concerned with scheduling the maintenance times for the cars, integrating the upgrade into the scheduled maintenance activity and allocating appropriate personnel, skill sets and resources to perform the upgrade installation simultaneously with the scheduled maintenance.

The supplier of the upgrade components is concerned with procuring parts and delivering them in time for the upgrade. The insurance company wants to make sure the components are tested thoroughly and present no safety issues. And the PM is responsible for stitching together the entire process. However, this task may not be as straightforward as portrayed. Ideally, interests would all be supportive of the end goal. But sometimes one individual interest in a group of interests may conflict with another. This represents the chance for program leadership to strategically view how all parts fit together and provide guidance and goal alignment. In the case of the rental car upgrade program, rallying each of the stakeholders behind the common goal to ultimately provide an upgraded vehicle to the rental car company could help align interests.

But what happens if interests and goals can't be aligned? Occasionally, a PM may ID a stakeholder, develop an engagement plan and attempt to align goals only to discover they are incompatible. Resources are limited and program leaders can't continue to allocate funds and personnel to address outside organizations' interests if they can't be aligned with the goals of the project.

A PM may be driven to revisit the identification or engagement steps—possibly removing an organization from the stakeholder list or tailoring the engagement approach. In a constrained environment, program leadership needs to evaluate if these "perceived" stakeholders indeed still add value and if the cost of focusing resources to address their issues is outweighed by benefit to the program. This section of the IDEA model enables reflection and forces routine reassessment of stakeholders, plans to engage and communicate with them, and the feasibility (or lack thereof) of goal alignment.

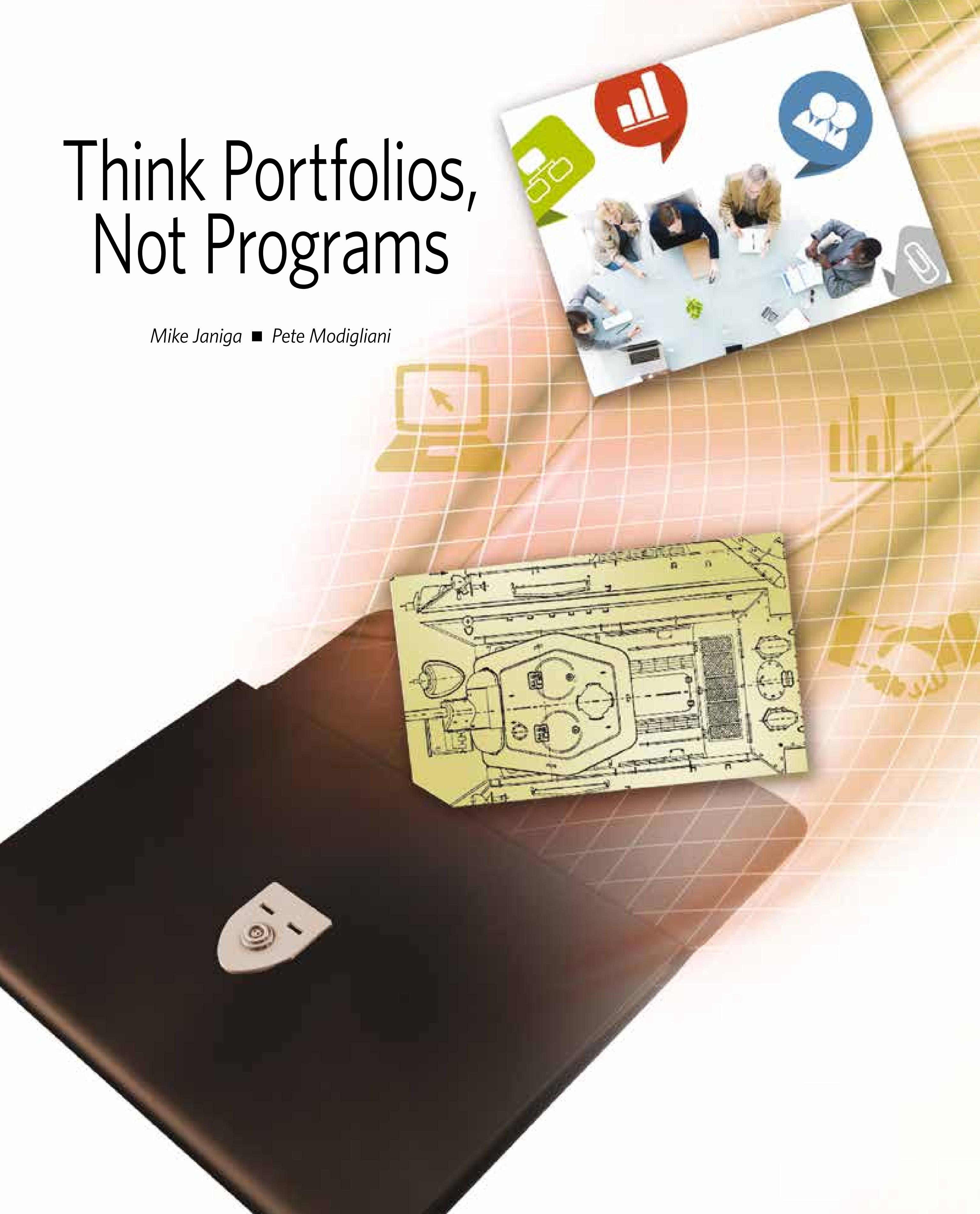
In conclusion, the IDEA model provides a suggested framework for PMs and leaders to address and synergize with stakeholders. First, PMs develop their criteria for defining stakeholders; then they identify specific organizations that fit this definition. Second, an engagement plan is developed to enable communication with the stakeholders, allowing PMs and leaders to gain a good grasp of stakeholder concerns. Finally, the PM must provide strategic guidance and direction to align all stakeholders with a common goal. Completing the cycle, program leaders should leverage lessons learned to continually reassess stakeholder identification and tailor engagement plans and feasibility for aligning interests and program goals.

In a dynamic environment, where resources are increasingly constrained, this ability to adjust an approach to stakeholder management gives program leaders and managers the flexibility to thrive in a variety of scenarios and ultimately to ensure acquisition program success.

The authors can be contacted at **franklin.gaillard@hanscom.af.mil** *and* **fgaillard@troy.edu**.

Think Portfolios, Not Programs

Mike Janiga ■ Pete Modigliani



The Department of Defense (DoD) can foster dynamic and innovative solutions for tomorrow's warfighter by designing acquisition portfolios that deliver an integrated suite of capabilities. Program executive officers (PEOs) today often focus on executing a dozen similar, but independent, programs. In contrast, large commercial businesses manage integrated product lines for items ranging from automobiles and electronics to software and health services. The DoD could leverage this model as a basis for constructing portfolios of similar programs that deliver enhanced capabilities in shorter timeframes.

Commercial Product Lines

Many large corporations organize their profit centers along product lines based on a successful product and fill out the line with associated spinoff products. For instance, Microsoft's well-known Xbox product line includes game machines, individual games and online services and apps. This linkage adds value for the customer and encourages further adoption of the full suite of products.

Companies designate a product line manager to maximize revenue and/or profit from the company's investment. To achieve this, executives provide significant latitude to product line managers to shape the product lines they manage—and that latitude includes marketing, new product development, forming corporate partnerships, research and development. Critical to the success of a product line is the ability to track the market closely and react swiftly to emerging trends and changes in consumer tastes before competitors do. Product line managers who perform these tasks effectively receive handsome rewards; those who do not do so quickly find themselves in a new line of business.

Breaking From the Program-Centric Model

In today's Defense Acquisition System, each program navigates the acquisition life cycle independently. Initial conceptual requirements drive program budgets, scope and solution space. Acquisition programs design, develop, test and produce individual systems that meet a defined set of requirements within an allocated budget.

However, today's complex and ever-changing defense environment requires integrated systems and services to produce capabilities greater than the sum of the individual parts. Analyzing alternatives and making trade-off decisions

Janiga *is the MITRE Corporation's senior acquisition executive and is responsible for the acquisition work program strategy across all of MITRE's federally funded research and development centers.* **Modigliani** *is MITRE's acquisition innovation research lead.*

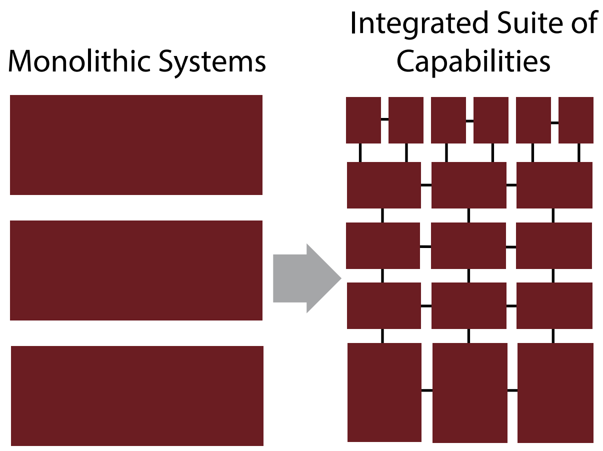
at the broader enterprise level rather than the program level would provide an opportunity to optimize performance, costs and/or risks. Guiding large systems independently through the acquisition life cycle over a period of 10 to 20 years has proven inefficient. The DoD can vastly improve the performance and outcomes of its acquisition system by incrementally delivering integrated capabilities via acquisition portfolios that feature tailored processes.

Just as industry constructs product lines, the DoD can structure acquisition portfolios around the system-of-systems concept. Each portfolio may include some or all of the programs in the current PEO portfolios or may be structured around another logical grouping of capabilities. As shown in Figure 1, a portfolio could decompose large systems into multiple smaller programs, projects or increments, and group those that contain similar capabilities, commercial off-the-shelf products, and services. For example, an IT portfolio for command and control or logistics could develop a suite of applications and services that run on a common infrastructure platform. Aircraft portfolios could be based on a common airframe (e.g., C-130) with different payloads, or on different airframes using common subsystems such as engines, communication suites or avionics software (e.g., Special Operations helicopters). This approach would not require a new top-down-driven structure; PEOs could start today by grouping a few related programs and tailoring a structure and process for increased efficiencies. The DoD could scale up these initial efforts after demonstrated success.

Solutions Analysis, Program Design

Conventional acquisition processes demand that programs develop and approve system requirements documents to meet the acquisition milestones. Under the recommended construct, the Initial Capabilities Document (ICD) should cover a broader mission or capability area and align with the scope of a portfolio rather than a program. Rather than

Figure 1. Decomposed Monolithic Systems Managed as an Integrated Portfolio



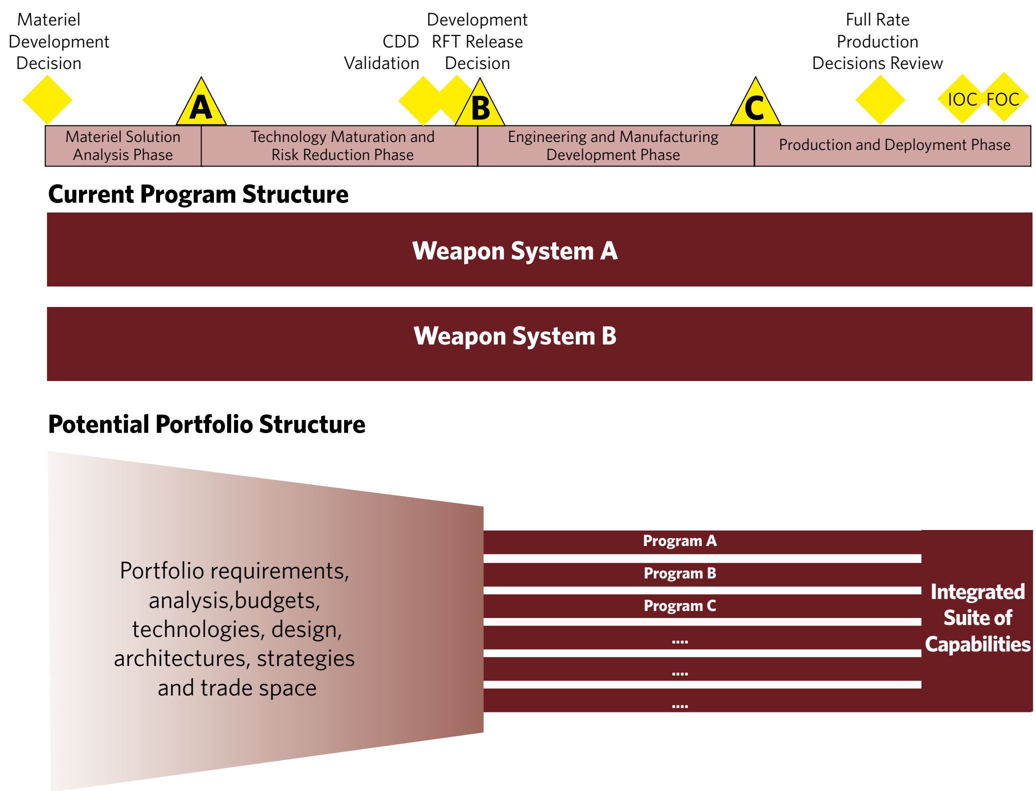
function purely as a milestone deliverable, the ICD should be a living document that operational sponsors could use to capture their current concepts of operations and prioritize a list of requirements in a database. An analysis of alternatives would no longer be a one-time event for a single system but would instead become a robust, continual process for optimizing the performance and/or efficiency of a portfolio of programs. These analyses would continuously monitor and evaluate a variety of technologies, systems, services and nonmaterial considerations such as doctrine, training or procedures. Advances in technologies could drive requirements changes and the resulting system capabilities.

According to current policies, the technology maturity phase focuses on prototyping and then perfecting the technology for the upcoming engineering and manufacturing development phase. The DoD increasingly relies on commercial technologies, and establishing a portfolio-level environment for technology development would enable a broader focus across increments and programs. It also would enable industry and government research and development (R&D) labs, centers and agencies to collaborate on an ongoing basis, conducting R&D funded by both government and industry. They could demonstrate capabilities, prototype emerging technologies, integrate existing capabilities and even compete in challenges. This would expand upon the development environments managed by the Service and agency R&D commands. As former Defense Acquisition Executive Dr. Jacques Gansler notes, "Military advantage will flow to those nations who can incorporate [commercial] technologies and practices rapidly into new systems and operations."

Portfolios could more effectively design the modular open systems strongly advocated by Congress, the Government Accountability Office, and DoD's Better Buying Power initiative. Collaboratively developed and proven standards, interfaces and processes would guide each program's development. Portfolio systems engineers would develop notional designs for each acquisition program using mature technologies from its development environment to address the top capability gaps identified in the relevant ICD. Robust portfolio enterprise architectures and notional designs would outline how each capability fits within the portfolio suite. Designing enterprise-level technical and business architectures would optimize portfolio performance over the program-centric designs used today. Portfolios should resist overengineering complex architectures by driving simplicity and making maximum use of commercial technologies.

The early phases of a traditional program could instead have a broader aperture in a portfolio approach, opening up the potential solution space (see Figure 2). As envisioned, acquisition programs would be smaller than the programs used for today's major systems, scoped in three- to five-year development increments. Smaller programs carry lower risk, as they simplify design, cost and schedule estimates—and ultimately delivery. Once managers effectively scope

Figure 2. Current vs. Potential Structures



a program, operational and acquisition stakeholders develop and approve a subordinate set of requirements and acquisition documents. For example, the IT Box concept in Joint Staff requirements policies features streamlined processes that focus on reducing the time taken to deliver software programs.

Portfolio Contracting

Contracting today involves a set of lengthy processes, with source selections that too often take a year or more to complete. The contractor or contractor team selected for the design and development of a new system often achieves monopolistic power over the government for a majority of a program's life span. As the DoD has moved toward acquiring larger and fewer major systems, this has changed the dynamics of the defense industry. Instead of creating a steady pipeline of potential work through periodic competitions, many of these large contracts become all-or-nothing, make-or-break outcomes that shape a major market segment for a decade or longer.

Competition remains the best way to drive down costs and increase innovation in defense programs. Therefore, a portfolio strategy should actively foster continuous competition over a program's life cycle via broad industry participation. Decomposing large systems into a smaller set of programs would increase opportunities for industry, especially small businesses, to compete for DoD work. A potential portfolio contract strategy could use multiple-award, Indefinite Delivery/Indefinite Quantity (IDIQ) contracts to establish targeted pools of large and small businesses with key technological and domain expertise.

The DoD could streamline contract timelines by establishing portfolio contracts with standardized business practices and precompeted contract vehicles to enable rapid generation of task orders for programs and program increments. These standardized business practices would include pricing, terms and conditions, templates and selection criteria. Continuous competition would be maintained by restricting the size of the contract vehicles with on and off ramps to refresh the vendor pools. Past performance on task orders within the portfolio also would represent a valuable selection criterion for future work as it would reward superior performance by contractors.

Portfolio Execution

A portfolio, once fully operational, would incorporate a robust suite of fielded capabilities, technologies in development and programs in the pipeline. A portfolio roadmap would provide strategic planning of the individual elements. Portfolio managers, like commercial product-line managers, could explore multiple alternatives to meet portfolio requirements by funding design and possibly development of a few small programs. The program demonstrating the best value in performance, capabilities, costs, schedule and risk management would receive funds for production. Those not selected could return to the portfolio development environment. Competition among programs would incentivize contractors to deliver their best performance on each program and spur government personnel to devise innovative strategies and solutions.

Portfolio strategies would focus on enterprise-level aspects, including defense industry considerations and major capital investments that resemble production lines. Portfolios could drive their programs to employ consistent, rigorous systems engineering and test processes detailed in sets of portfolio documents. Software for managing project portfolios would

integrate program schedules to show dependencies and impacts of schedule slips, budget cuts or other scenario planning events. Programs would report a common set of metrics to give managers a holistic view of portfolio health.

Dynamic Resource Allocation

One of the biggest challenges in implementing a portfolio structure concerns the allocation of program budgets. Most programs today are funded via accounts called program elements (PEs). Transferring funds between PEs requires lengthy approvals by senior DoD officials and possibly by Congress. However, some PEs include multiple programs, with each broken out at a subaccount level called a budget program activity code (BPAC). Transferring funds between BPACs requires lower approval thresholds. Thus, allocating a portfolio budget at the PE level would offer funding flexibility and agility, while also providing sufficient transparency to oversight officials.

This funding approach would increase the effective use of constrained resources and direct funds toward the highest-priority capabilities with the greatest enterprise impact. Pentagon executives would focus on strategic budget allocations at the portfolio level. Portfolio stakeholders would allocate program funding following key milestone reviews. Portfolio managers would establish funding lines for technology development, enterprise platforms and personnel for enterprise efficiencies. Fortunately, such a change would not require a wholesale restructuring of the planning, programming, budgeting and execution process but simply would call for shaping a few PEs for an initial set of portfolios.

Portfolios also would provide an opportunity to make better use of staff by developing subject matter experts and dynamically assigning them across the portfolio programs. Experience is critical for complex system acquisition, yet today roughly half of DoD's acquisition workforce has less than five years of experience. Sharing staff across multiple programs in a portfolio would help junior staff gain a deeper knowledge base across a diverse set of programs.

Designing Acquisition Portfolios

The principles of authority, simplicity, commonality and agility should guide all acquisition portfolios. By adopting the commercial product-line approach, the DoD would address long-standing issues associated with acquisition speed, agility and system interoperability. Elevating the time-consuming acquisition processes to the portfolio level would reduce program workload, allowing each program to deliver products faster.

In a complex, integrated environment, the Defense Acquisition System can no longer rely on a structure based on individual systems but rather should embrace a capability-focused, portfolio-centric structure modeled on the commercial sector. Managing requirements, budgets and staffs at the portfolio level would enable dynamic allocation to high-priority programs. Portfolio strategies, roadmaps and architectures would guide program development.

An active government and industry portfolio community would collaboratively develop technologies and designs and employ continuous competition to develop and produce the individual programs. Portfolios would design and optimize acquisition processes to deliver a suite of smaller programs rapidly, ensuring that warfighters regularly receive incremental capabilities that incorporate the latest technologies designed to achieve their operational missions.

Apple did not revolutionize consumer electronics because the iPod outperformed MP3 players. Instead, integrating the iPod with iTunes proved the critical differentiator and led to the iPhone and iPad. Toyota does not design, develop and produce the Camry without considering the Corolla, Prius and other models but creates technologies for hybrids and electric vehicles and integrates the innovations across the product line. Similarly, the DoD and its industry partners can organize around portfolios of capabilities that extend beyond a single system, while regularly delivering smaller increments of functionality—equivalent to a particular car model that shares many features of the broader product line. In this way, portfolios would enable strategic cost efficiencies in budget-constrained environments while improving effective tactical response for current operations.

The authors can be contacted at **mjaniga@mitre.org** *and* **pmodigliani@mitre.org**.

Mining Hidden Gems

Extract Information Systems' Value

John Kruse ■ Maura Slattery

More than 10 years ago, the Department of Defense (DoD) Chief Information Officer took a bold step toward broad information sharing by publishing the seminal Net-Centric Data Strategy. Since then, the Services have made great strides by creating many new data sources across the DoD. Still, taking advantage of all this pent-up capability and value remains a difficult task for most of the enterprise.

The data or capabilities within any given program of record (PoR) system may be valuable to others, both known and unanticipated, but often there is little understanding of how we might extract this value or how mining our existing resources might change the way we do business.

This value often can be exposed quickly and at low cost. Nevertheless, Enterprise Integration (EI), the activity that stitches together disparate systems and data, is not well understood or utilized as often as might be warranted. Some of this is because of systemic issues within DoD acquisition, but much of it is due to a perception that EI is big, expensive and high-risk. In short, there is very little recognition within PoRs that the rewards of EI can outweigh its costs and risks. This article outlines how the Air Force's C2 Constellation program found a successful

Kruse, *Ph.D., is lead information systems engineer at MITRE Corporation in Bedford, Mass.* **Slattery** *is principal multi-discipline systems engineer at MITRE.*

approach to EI by carefully selecting initiatives that are aligned with PoR plans and that are supported by warfighters.

Over time, however, it became clear that trying to promote EI from the top wasn't having the anticipated impact but that smaller, more focused efforts seemed to get better traction.

C2 Constellation

Since 2001, the Command and Control Constellation (C2 Constellation) program has been the "sole Air Force program for defining, developing and assessing integration of global, theater and tactical level Air Force air, space and cyber C2 capabilities in support of the joint warfighter." Until four years ago, the program tried to span EI. It focused on creating enterprise architectures (EAs) to help "Big Air Force" drive acquisition and systems engineering decisions while attempting to effect specific changes with focused EI projects. An underlying assumption behind the program's top-down efforts was that if someone could simply identify and document smart choices for systems engineering in support of EI, programs could adopt these suggestions and the enterprise would benefit. Over time, however, it became clear that trying to promote EI from the top wasn't having the anticipated impact but that smaller, more focused, efforts seemed to get better traction.

Why Top Down is Difficult

Creating EAs makes a lot of sense. Rather than have PoRs building systems haphazardly with only their own immediate requirements in mind, we should seek ways to standardize and create rational, repeatable patterns that can provide efficiencies in development, integration and operation. However, in order to provide real value and efficiency, EAs need to achieve a critical mass of adoption, and in our current acquisition environment it is difficult to achieve this across a broad and heterogeneous enterprise.

"To be" EAs are by definition top-down and conceptual in nature. To provide value, they require that (1) an acceptable standard architecture can be accurately defined, and (2) that once defined, we can realistically propagate the architecture among the PoRs to realize its benefits. Even when we achieve the first requirement, our decentralized acquisition system makes it very difficult to achieve the second.

EAs may fail because they are poorly conceived, but far more often they fall prey to an acquisition environment that does not reward cross-PoR cooperation and standardization. PoRs are funded, incentivized and judged by how they deliver capabilities in response to a specific set of requirements for a specific set of warfighters. If a PoR fails to provide benefit to its core set of users, the program is by definition a failure. Thus, conforming to enterprise-level architectures or standards that address the needs of a broader community often is reasonably met with, "We don't have a requirement for that."

C2 Constellation faced such a situation in which the programs and portfolios with which it worked were not, for a number of reasons, willing to implement the developed EAs. As a result, C2 Constellation's leadership decided to revisit its broad-front EI approach. Rather than pursue a strategy that emphasized top-down efforts, the program shifted focus to building bottom-up integration bridges among those who were keen to achieve particular tactical ends. These changes could in turn be leveraged to help the broader enterprise. Thus, by helping PoRs meet their specific, documented requirements faster and at lower cost, the whole enterprise could benefit. Since it has shifted its emphasis, C2 Constellation has enjoyed greater impact with PoRs, and for surprising and simple reasons that might have implications for broader information technology (IT) acquisition.

Factors Influencing Success

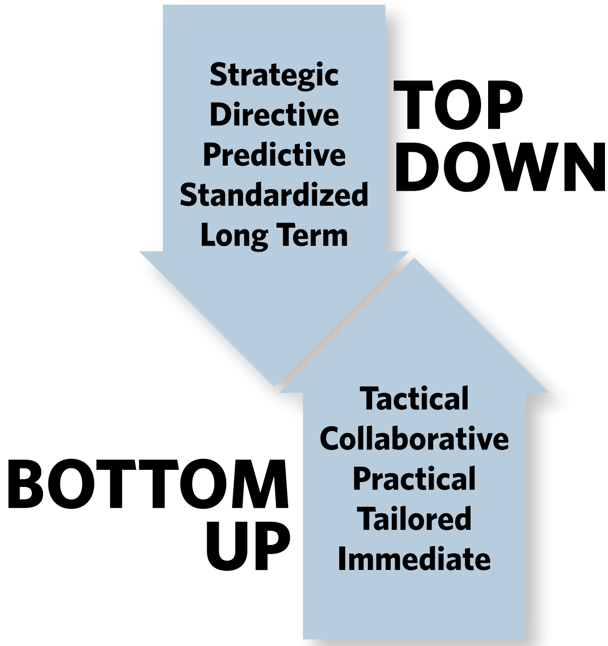
Any discussion about IT program or project success would need to acknowledge that a wide range of factors influence success and that there are many potential pitfalls from the genesis of an idea to successful transition. In our experience, however, beyond the standard concerns of performance, cost and schedule, EI issues can generally be simplified into three interrelated classes involving risk. The first two have to do with a capability's alignment with the two major stakeholders—the warfighters and the PoR—throughout the EI effort. The third, limited complexity, involves the ability of the PoR to limit risk through timely delivery of effective capabilities, given complicated technical and operational landscapes. The following are brief explanations of each of the three and how C2 Constellation realized that they come to influence success.

Operational Community Commitment

The No. 1 question we must address when pursuing EI is, "Who is asking for this?" All the varied stakeholders should have a say in acquisition decisions, but the warfighters should take priority. Without their backing, transition may be technically achievable but may never attain its intended ends. This is especially common when initiatives cross system or organizational boundaries, as is common in EI.

There are good reasons why warfighters tend to be eager to experiment with new ideas but are much more selective about what actually moves forward to transition. They are best placed to imagine the ripple effects and potential risks in everything from training to sustainment that comes with

Figure 1. Top-Down vs. Bottom-Up Enterprise Integration



a new technology. Additionally, there are significant barriers to any technology that changes the way the organization operates. In the words of Rear Adm. Tom Zelibor, the Navy's fleet commander during Operation Enduring Freedom and a technology pioneer, "I've always maintained that the hardest part of this isn't the technology, it's the culture." Technologists and program managers may understand many things, but we are not the people who can make accurate calls about how much change a command is willing to assume or the true net worth of a new capability within a greater operational context.

PoR Alignment

The second place where we see new innovations and initiatives fail is in their simple nonalignment with the PoRs in terms of established technical architectures, functionality, acquisition strategy or timing for a smooth technology transition. Expecting them to make even seemingly simple accommodations in transitioning capabilities is often unrealistic within the cost, schedule and performance constraints of the program.

In part because they are cross-organizational, bringing an EI innovation or initiative to fruition in the field is akin to running a gantlet where any single issue might stop an initiative in its tracks or sap its ability to get over the next hurdle. Often, these issues have nothing to do with the wishes of the warfighters, the developers or the participating PoRs. For instance, one common problem in transitioning EI innovations is that of cycle-time mismatches in which a PoR is interested but is simply not ready for the innovation as it already has committed its time and resources. Delay may be possible, but frequently the developers and other PoRs must move on to new work, which often involves disbanding the effort. In such situations, it is difficult to revive stalled initiatives—and, when momentum is lost, even great ideas tend to wither.

Limited Complexity

Once we have moved beyond the organizational and social needs for warfighter commitment and PoR alignment, we must deal with the elusive problem of limiting complexity. Under conditions of great complexity, our abilities to understand systems, extract good requirements and develop compelling capabilities begin to fail. Heightened complexity often leads to either analysis paralysis—in which we are unable to decide what to do—or slow and difficult acquisition that misses the mark and underwhelms the end users.

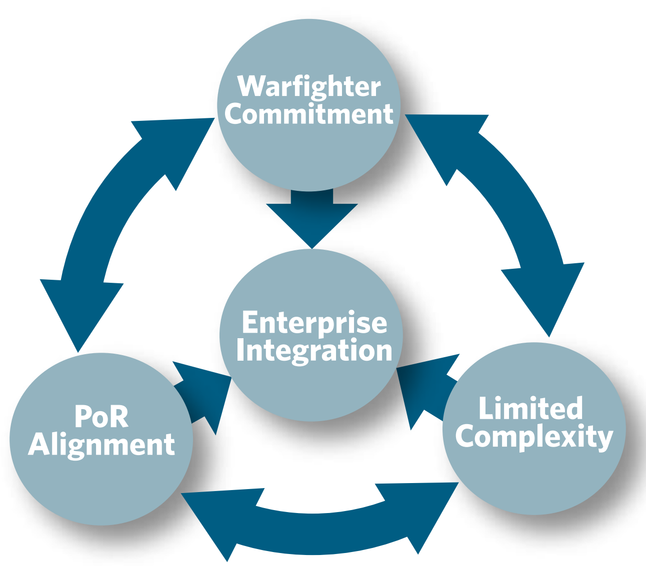
Moreover, highly complex EI initiatives can increase downstream risks as they have implications for acceptance, security, training and maintenance. Typically, the relationship between system complexity and technical difficulty is not linear—that is, as complexity increases, the associated technical difficulties and risks compound even faster. Thus, a complex EI solution can either be difficult to transition or it may be limited in operational use.

Recipe for Success

When C2 Constellation changed its approach to EI, the program was simply trying to find commonsense ways to identify valuable opportunities, develop them and then transition improvements to the field. The program decided to work directly with interested PoRs to find targeted EI solutions and then provide relatively modest funding to perform the work and some engineering and project management support to help the process along.

As a result of our own particular environment, and previous experiences, C2 Constellation's leadership explicitly set several criteria for selecting new EI projects that were intended to

Figure 2. Enterprise Integration Factors



maximize the chances for successful development and transition to PoRs. Every initiative had to be submitted within a focus area as defined by our sister organization, the Air Force Command and Control Integration Center (AFC2IC). Because the focus areas could change from year to year, all projects' proposals would need to produce a valuable product at the end of each fiscal year, as specific focus areas might not be continued. Beyond this, projects were specifically evaluated in terms of (1) warfighter impact, (2) transition likelihood and (3) cost.

Technologists and program managers may understand many things, but we are not the people who can make accurate calls about how much change a command is willing to assume.

In retrospect, it is easy to see that our measure of warfighter impact stood in relatively well as a measure of warfighter commitment. Typically, if a given initiative was expected to have high end-user value, the warfighters would show commitment and even enthusiasm. But as mentioned previously, this support was based on their holistic evaluation of the pros and cons of actually using the EI innovation.

Similarly, our transition likelihood assessment was a reasonable metric for the many facets of PoR alignment. By asking the PoRs for an opinion on this likelihood, we were getting their opinion on how well the initiative was aligned with their current and planned architectures and states. They simply did not want to invest time or resources in any effort that was unlikely to help them deliver capabilities to the warfighter.

In retrospect, we found we had been limiting initiative complexity with our one-year focus and our limited budgets. Everyone understood that cost and schedule were effectively fixed and, if we could not produce something valuable within these constraints, the effort would never be extended—much less transition to a PoR. This tended to lower the tolerance for risk and consequently limited complexity as the stakeholders wanted crisp, understandable and achievable initiatives. Additionally, modest initiatives are less likely to violate organizational culture and norms, which can help gain acceptance and successful transition.

A telling example of this approach would be the Integrated Tactical Airspace (ITA) initiative that sought to knit together Army and Air Force tactical systems to dynamically share airspace data in support of more Agile and coordinated operations. This collaborative effort involved three Army and one Air Force systems sharing airspaces through a community-defined data standard. The initiative had PoR alignment that was cemented by resource sharing among the joint participants. Both the Army and Air Force users were committed to the initiative as they were anxious to finally have a capability that could support the operational vision that had been established.

Finally, the technical complexity of the effort was controlled through the use of the common data standard and a modest, modular development approach. As a result, the developed prototypes are being moved into the baselines of the respective PoRs.

The Bottom Line

The new bottom-up EI approach has greatly improved the effectiveness of C2 Constellation and the value proposition that we offer to the PoRs and the warfighters. Even in cases where a direct transition to the warfighter was unachievable, it was often possible to affect the PoRs positively through new/changed requirements, improved data schemas, etc. In a recent study of initiative outcomes over the last three years, we found that 16 of 19 (or 84 percent) of our speculative initiatives bore fruit.

A positive secondary effect of the new EI approach was the emergence of resource pooling to achieve results. PoRs are willing to contribute substantial time and complementary resources, and this contribution then cements a high level of commitment to the team effort. The warfighters, in turn, have been positive about collaborating on crosscutting capabilities. Embarking on this approach can form the basis of a virtuous cycle in which all of the various stakeholders come together.

We believe that, if more widely pursued, this EI approach has potential in efficiently tackling cross-PoR requirements. Furthermore, our findings about the benefits of limiting complexity with short schedules may have real merit for the efforts of more conventional PoRs. When one limits an effort to one year, it automatically changes the assumptions, focuses effort and lowers risks. The relationship between the time allotted to an IT project and the chance that it will not meet expectations has been noted in the commercial world—"the longer a project is scheduled to last, the more likely it is that it will run over time and budget, with every additional year spent on the project increasing cost overruns by 15 percent," according to a McKinsey and Company report. There also are signs that the U.S. Government already is shifting toward using shorter development cycles as a means for improvement. As Roger Baker, chief information officer of the Veterans Administration, said, "We are huge fans of Agile [development], and are using it in our most critical programs."

The authors can be contacted at **wkruse@mitre.org** *and* **slattery@mitre.org**.

We're Looking for a Few Good Authors

Got opinions to air?
Interested in passing on lessons learned from your project or program?
Willing to share your expertise with the acquisition community?
Want to help change the way DoD does business?

Write an article (ideally, 1,500 to 2,500 words long) and *Defense AT&L* will consider it for publication. Our readers are interested in real-life, hands-on experiences that will help them expand their knowledge and do their jobs better.

What's In It for You?

First off, seeing your name in print is quite a kick. But more than that, publishing in award-winning *Defense AT&L* can help advance your career. One of our authors has even been offered jobs on the basis of articles written for the magazine.

Now we can't promise you a new job, but many of our authors:

- Earn continuous learning points
- Gain recognition as subject matter experts
- Are invited to speak at conferences or symposia
- Get promoted or rewarded

For more information and advice on how to submit your manuscript, check the writer's guidelines on the inside of the back cover of this issue or at http://www.dau.mil/publications/ATLdocs/Writer's%20Guidelines.pdf or contact the managing editor at datl@dau.mil.

If you're interested in having longer, scholarly articles considered for publication in the *Defense Acquisition Review Journal*, or if you're a subject matter expert and would be willing to referee articles, contact the managing editor at defensearj@dau.mil. Be sure to check the guidelines for authors at http://www.dau.mil/publications/DefenseARJ/ARJ/ARJ70/ARJ-70_GuidelinesforContributors.pdf.

The Path to Software Cost Control

Dr. James R. Eckardt ■ *Timothy L. Davis* ■ *Richard A. Stern*
Dr. Cindy S. Wong ■ *Richard K. Marymee* ■ *Arde L. Bedjanian*

Many programs risk cost growth and schedule delays because of software development issues. In the 2010 Government Accountability Office (GAO) defense acquisition report, *Assessments of Selected Weapon Programs,* the programs with count growth in significant source line of code (SLOC) since development startup experienced accelerated cost increases and excessive schedule delays relative to other programs. The report asserted that collecting, tracking and containing software defects in the phase where they occur is an excellent cost-control management practice. Programs surveyed indicated that an average of 31 percent of defects corrected were detected after the development phase in which they were inserted. Capturing software defects in phase is critical because detecting defects out of phase results in expensive program rework.

Real Cost Impact

Software defects are so prevalent and detrimental that they cost the U.S. economy an estimated $59.5 billion annually, or about 0.6 percent of the gross domestic product, according to a 2002 study commissioned by the Department of Commerce's National Institute of Standards and Technology (NIST). A more recent Cambridge University study reported that the global cost of debugging software has risen to $312 billion annually. The research found that, on average, software developers spend 50 percent of their programming time detecting and fixing defects.

Bedjanian *is founder and president of GreenDart Inc., a San Pedro, Calif., firm focused on identifying software issues early in development cycles.* **Davis**, **Stern**, **Eckardt**, **Wong** *and* **Marymee** *are systems engineers at GreenDart.*

Some Recognized Challenges

Defect-removal efforts substantially compound several parallel factors that also result in significant program cost growth:

- **Decreased Product Life Expectancy**—Due to technology advances and rapid product evolution, the life expectancy of software products has decreased dramatically over the past several years.
- **Increased Program Complexity**—The size of software products no longer is measured in thousands of lines of code but in millions.
- **Optimistic Software Reuse Plans**—Many programs propose aggressive software reuse in order to lower the proposed cost of the software without reducing the estimated software size.
- **Requirements Growth**—A current trend toward "late binding" along with the revision of customer requirements during development risks an introduction of an unintended requirements creep. This disrupts predevelopment cost and schedule estimates.
- **Curtailed Testing**—As development progresses, many programs experience a cost growth and schedule slip that result in a simplified "back-end" testing agenda to recover some schedule. This approach emphasizes test for success (verifying all requirements are met) and limits test for failure (the search for critical flaws).

These factors place early pressure on developers to maintain schedule commitments, leading to increased reliance on final product testing for defect detection. In the commercial realm, the increased use of "beta releases" is a symptom of this. However, studies have shown that optimal schedule and cost outcomes actually occur with rigorous early detection and removal of defects. This paper presents a means to move toward that optimum.

The Software Development Life Cycles (SDLC) adheres to critical phases that are essential for product development. These phases include planning, analysis, design and implementation and may include concurrent system evaluation, information gathering and feasibility studies. Traditional waterfall SDLC may be replaced by variations of the Agile/SCRUM (the later involves multiple small development teams) development methodology, due in part to today's increased program complexity and module count. No matter which process is implemented, defect insertion can occur during the correction of the identified defect and will additionally impact program cost and schedule.

Typical Defects and Frequency

Reference data indicate that about 40 percent of defects originate in the requirements definition phase (with design accounting for 10 percent, code for 45 percent, and test for

Table 1. Typical Software Development Life Cycles (SDLC) Phase-Related Defects

SDLC Phase	Typical Defect
Requirements Definition	• Requirements, and associated data, are not traced correctly, are missing or aren't stated clearly. • Software requirements specifications, interface requirements specifications, test approaches/data, algorithms are incorrect and/or inconsistent. • Inadequate and/or incorrect user interface as input from user groups.
Design	• Incorrect or inconsistent interface traceability between documents. • Requirements are not satisfied by the software design. • Critical functions and/or algorithms have been identified but not correctly described. • Design risk and risk mitigations have been incorrectly identified.
Code	• Incomplete source code, unused or unreachable code. • Incorporation of "buggy" reuse code and ineffective integration of commercial off-the-shelf (COTS) and government-furnished equipment (GFE) software. • Failure to track code corrections, uncompleted code and code-completion schedules. • Failure to systematically identify critical and hazardous components of the code for additional risk management. • Inadequate/incorrect/misleading or missing comments in the source code. • Standards and project-related design/requirements/coding standards not followed.
Test	• Failure to track code corrections, incomplete code and code-completion testing schedules. • Failure to ensure that hazardous and otherwise critical components of the code are thoroughly tested. • Limited test data used in component development and testing. • Incomplete developer test plans, test procedures or test execution results. • Limited testing and review of results do not adequately demonstrate that the software supports mission requirements and capabilities.

... More than a third of these costs ... could be eliminated by a more rigorous software assessment process that enables earlier and more effective detection and correction of software defects.

5 percent). Of these defects, the requirements phase only detects and corrects about 15 percent (design corrects 10 percent, code 45 percent and test 30 percent). Table 1 depicts a list of typical phase-related defects independent of SDLC process model used.

Cost of Latent (Out-of-Phase) Defects

Defects not removed in their respective creation phase are subject to a substantial—and escalating—repair cost penalty when corrected later. For example, a requirement defect detected in operations resulted in a cost 368 times greater than it should have been, according to NASA's study of return on investment (ROI) for software independent verification and validation (IV&V). Delayed defect correction increases rework (cost/schedule impact) required to correct the defect. Delayed defect correction typically involves making numerous changes to both the original and now related software, to intermediate work products (such as test procedures) and more extensive regression testing. More change activity also increases the opportunity to introduce new defects during the delayed corrections.

Figure 1, Latent Defect Cost Escalation, compiled from this NASA study illustrates the relative cost escalation of correcting an out-of-phase defect. In this figure, an in-phase corrected defect receives no cost impact but, if the detection and correction occur in a subsequent phase, the costs increase exponentially. This cost penalty creates a great incentive to identify and correct the defect in phase.

According to the 2002 NIST study, not all defects can be corrected in a cost-effective time span. However, more than a third of these costs, or an estimated $22.2 billion, could be eliminated by a more rigorous software assessment process that would enable earlier and more effective detection and correction of software defects.

Addressing Developmental Program Latent Defects

Major cost savings at the total program level are achievable by systematically containing most software defects in or near the phases where they are introduced. Detecting latent defects as early as possible is best, specifically if corrected in the phase where they are introduced rather than detected later. Current defect-detection strategies include: (1) independent testing; (2) developer verification and validation (V&V); and (3) IV&V. As will be shown, only one of these approaches is effective for identifying potential latent defects within the phase where the defect is introduced.

Independent Tests to Identify System Defects

Independent testing brings significant value to the final acceptance of software systems. These tests typically are executed on completed systems by an organization (or separate company) independent of the development effort—which increases system assessment objectivity. The problem with addressing latent defect costs using this approach is timing—the testing occurs much too late in the SDLC to reduce latent defect impacts. Therefore, independent testing is not a mechanism for reducing latent defect costs.

Why an "Independent" Effort Is More Effective

Development organizations address V&V in two ways: (1) employing a product review process at the end of each phase of the development by the developers themselves; and (2) using a separate team to V&V the developed products. While developer V&V may encompass many forms of development

Figure 1. Latent Defect Cost Escalation

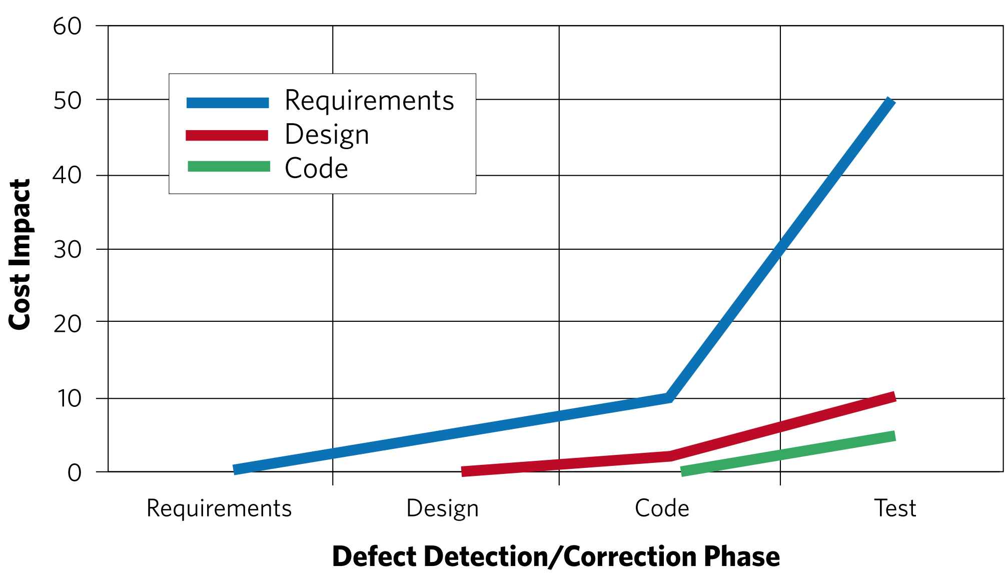
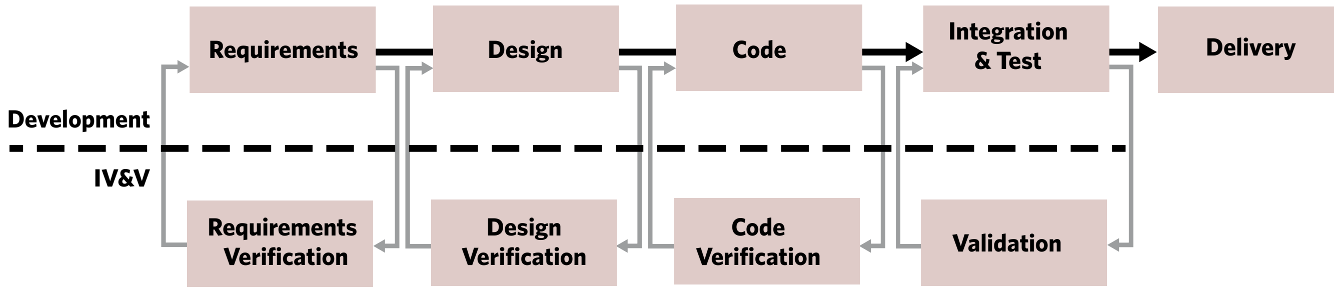


Figure 2. IV&V Process Tied to SDLC Phases



testing, the developer's primary focus is requirements sell-off "test for success" verification activities. However, a significant portion of the defects identified in Table 1 are not detectable by this strategy. To capture these types of defects, the approach must include a "test for failure" focus (e.g., limit checking, off-nominal condition analysis, etc.). These are not typical requirements sell-off strategies and, therefore, are not activities performed by the developer's V&V team. They are, however, key strategies of an effective IV&V effort.

Reducing the Latent Defect Impact

IV&V is a software assessment technique that integrates with the developer's process to capture, assess and report on defects in developed products. A sample IV&V program, linked to developer activities, shown in Figure 2, integrates the developer's waterfall SDLC process with IV&V assessments and feedback loop responses. The outputs for each developer phase are assessed, and feedback (e.g., identified defects) is provided to the development team in phase. IV&V maximizes

Table 2. IV&V Tasks to Eliminate Latent Defects

Requirements Verification	• Validate that the requirements are complete, concise, understandable, testable and that they satisfy the user's needs. • Verify that the developer requirements are traced accurately to software components and back to the system and interface requirements. • Evaluate risks associated with the requirements and with the concepts and plans for testing. • Review software requirements specifications, higher-level requirements and interface requirements specifications for consistency. • Ensure that test approaches and test data are correct and consistent. • Ensure algorithms are consistent with requirements and test planning and that the algorithm test plans are sufficient.
Design Verification	• Verify that the interfaces are correct and consistent in all documents. • Validate that the requirements are satisfactorily implemented in the design and that the design satisfies all of the requirements. • Review the reuse code and the reuse plan to ensure the feasibility of reuse as planned. • Verify that the critical functions and algorithms have been identified and prototyped and are addressed in the design. • Ensure that the developers have correctly identified design risk and security issues and appropriate mitigations. • Ensure that test procedures and test data are correct and consistent.
Code Verification	• Analyze supplied code with code analysis tool(s), identifying any code debug/violations. • Track code corrections, incomplete code and code completion schedules. • Ensure that critical and hazardous components of the code are identified. • Monitor code development performing design through code trace analysis. • Evaluate unit test artifacts for completeness, addressing relevant requirements and off-nominal testing.
Validation	• Validate that test results address the user's needs and system requirements. Validate test results against expected results in test plans. • Identify and track retest of corrections, incomplete testing, and retest/regression test completion schedules. • If developer cost and/or schedule overruns occur, identify and evaluate mitigation options.

development insight, identifies weaknesses, assesses failure conditions and uncovers defects as they are introduced into the system—thereby reducing the potential for latent defect propagation into later phases.

Other nonwaterfall developer processes (e.g., Agile, etc.) are also accommodated by an IV&V integration strategy.

The application of IV&V is unique to each development effort, based on such factors as customer's priorities (where to focus), developer strategy, developer processes and products and the application of IV&V tools unique to the particular development effort. Typical IV&V tasks include those listed in Table 2. When the tasks referenced in Table 2 are executed successfully, critical defect detections are accelerated, thereby saving program costs through minimized rework, reduced development schedule and decreased operational maintenance costs.

Identifying defects early and, hence, saving program costs requires an investment in IV&V tasking. So we come to the real question: "Is the price of the IV&V effort justified by the program cost savings?"

IV&V Return on Investment

In 2012, GreenDart, along with the NASA IV&V Facility in West Virginia, conducted a study into the long-term effects of IV&V on program development costs. Based upon the NASA-provided development and IV&V defect-identification information for 31 programs, the paper concluded the ROI for IV&V ranged from a conservative 85 percent to a maximum 294 percent above the cost of performing the IV&V.

Therefore, an investment in IV&V returns at least 85 percent program savings beyond the cost of the IV&V effort. In the most extreme cases, IV&V returned 294 percent program savings. In short, the investment in IV&V is justified.

Computed IV&V Cost Savings

The example in Figure 4 illustrates the impact of including IV&V in a software program's development. The results of the GreenDart-NASA and NASA IV&V ROI studies show that, for a program with an initial development cost of $90 million, latent defects are estimated to raise the project's actual cost to $115 million. The customer can reduce some of this cost by adding IV&V. Using the conservative ROI of 85 percent, the following calculation shows that $6 million spent on IV&V reduces the cost of latent errors by about $11 million:

Figure 3. IV&V Return

* Return on investment, excluding investment costs

($6 million IV&V) X 1.85 = $11 million latent defect savings.

Subtracting the cost of IV&V from this gives a software development savings (excluding schedule savings) of:

$11 million – $6 million = $5 million net savings.

It is important to note the final program costs are still in excess of the proposed $90 million price:

TOTAL COST: $90 million + $25 million (latent defects) – $11 million (latent defect savings) + $6 million (IV&V costs) = $110 million.

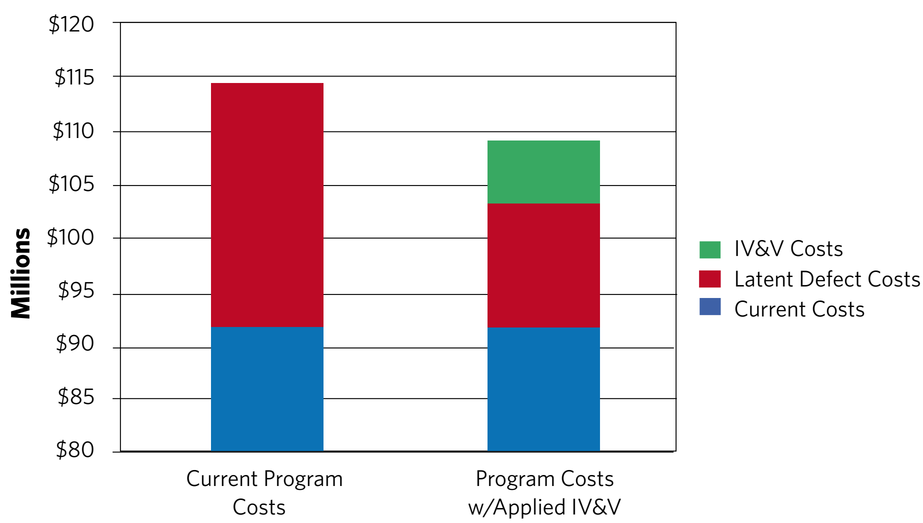
Additional program measures must be employed to sustain a $90 million cost profile (reduce program requirements, etc.).

Conclusion

Many factors contribute to software development cost overruns. One major cost impact is latent defects. Significant study results, presented in this paper, identify the latent defect cost impacts and the positive cost savings of an effective IV&V program.

The authors can be contacted through **arde.bedjanian@greendart.aero**.

Figure 4. Anticipated IV&V Results

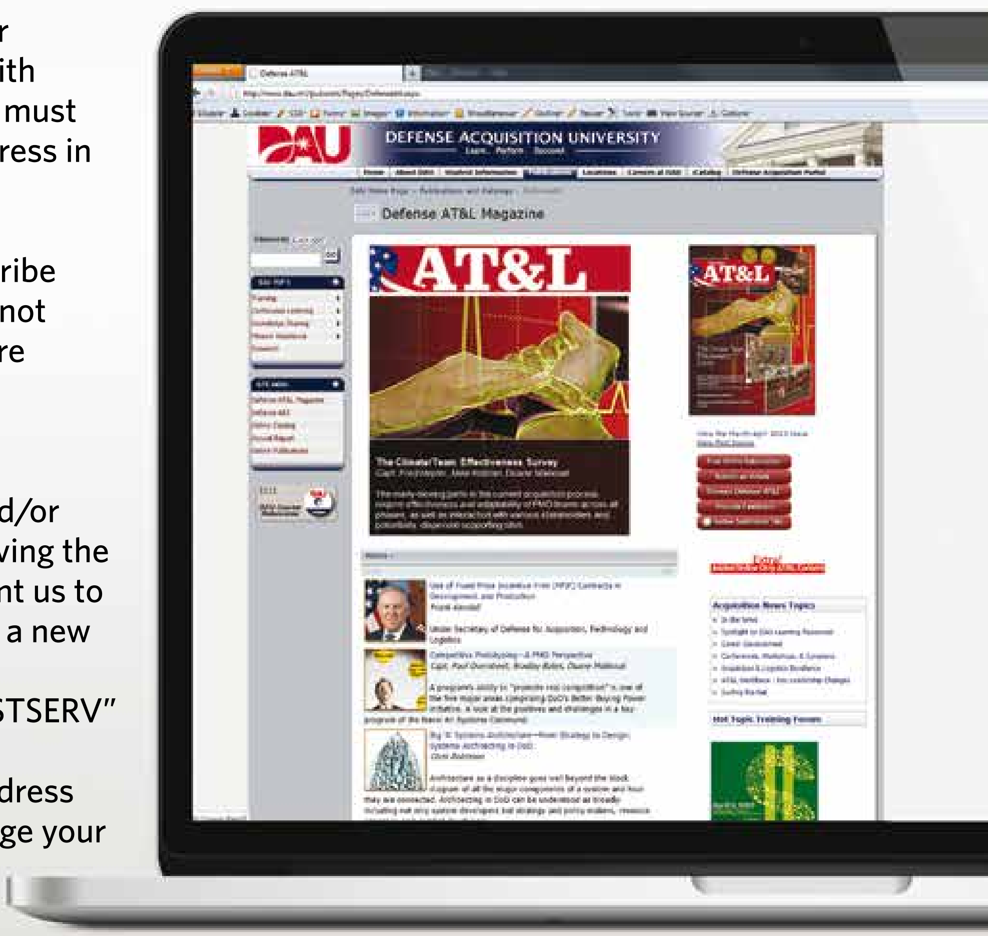


Defense ARJ and ***AT&L*** have become online-only publications for **individual subscribers**

If you would like to start or continue a membership with ***Defense ARJ*** or ***AT&L***, you must register a valid e-mail address in our **LISTSERV**

All Readers: Please subscribe or resubscribe so you will not miss out on receiving future publications.

- Send an e-mail to darjonline@dau.mil and/or datlonline@dau.mil, giving the e-mail address you want us to use to notify you when a new issue is posted.
- Please type "Add to LISTSERV" in the subject line.
- Please also use this address to notify us if you change your e-mail address.



http://www.dau.mil

Meaningful **Metrics**

Measuring Success of Software Integration Testing Labs

Christian Hagen ■ Steven Hurt ■ Andrew Williams

The U.S. military is moving from a world dominated by advanced hardware to one of fully integrated, complex systems of both hardware and software—a move that makes it even more relevant for the military to understand how to measure and test systems with data-driven metrics and easily measurable results.

Weapon systems program offices have developed full-system and subsystem integration laboratories with the primary mission of testing and certifying integrated hardware and software during the systems' development, modernization and sustainment. These labs play a critical role in delivering a war-winning software and hardware capability to the warfighter in the battlefield. As a result, these labs have become essential to our country's defense and support of our foreign policy.

Hagen *is a partner in A.T. Kearney's Strategic Information Technology Practice and is based in Chicago. He advises many of the world's largest organizations across multiple industries, including government and defense contractors.* **Hurt** *is a partner in A.T. Kearney's Public Sector and Defense Services Practice and is based in Dallas. He has worked with several of the U.S. Air Force's highest visibility programs to drive affordability in both software and hardware sustainment.* **Williams** *is a manager in A.T. Kearney's Dallas office and works with military programs to drive improvement and cost reduction in software engineering operations.*

However, each lab throughout the Department of Defense (DoD) has developed its own unique processes, specific to individual programs, for measuring its progress and success. This nonstandard, and often ad hoc, approach has caused confusion among DoD program leaders about what the metrics mean, which ones matter and how to make fully informed command decisions about the software integration labs.

Without meaningful metrics that can illuminate the labs' actual performance, program leaders are unable to make even minor decisions—let alone major ones—about running an individual software lab or groups of labs within the DoD. They simply cannot manage the labs effectively. Leaders can't answer questions about whether a lab is running cost-effectively, about what a lab's efficiency is or if that level of efficiency is good or bad, or about whether to send more or less work to a particular location. Moreover, military and software leaders don't know how much money to invest in updating a lab, whether it would be best to close a lab and move the testing somewhere else or even whether buying a new piece of equipment would reduce the lab's overall costs and improve its performance. Without an appropriate approach to software integration laboratory metrics, leaders are operating the labs in the dark with little visibility on whether their decisions improve or hurt development, sustainment and modernization.

Program leaders are now making decisions with the engineering- and technology-based metrics favored by those who are far more concerned with what's needed to test, say, a third-generation radar unit than with the cost, efficiency and performance of running a lab. They have no valid metrics relevant to those who must make command-level decisions from a holistic, business perspective. Having this information on testing productivity has never been more important. We need to look no further than the F-35 program, whose software has expanded to about 24,000 source lines of code (see Figure 1). The indications are that much of the F-35's well-publicized delays are the result of its inability to test software.

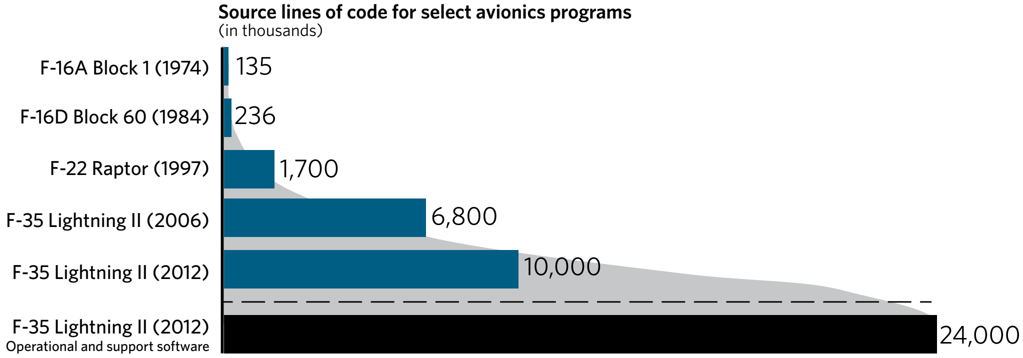
Recently, the leaders of a major DoD program tried to determine what the impact on its operations would be if they moved a specific lab to another geographic location. Because they had no standard set of metrics, a new approach would be needed to make a decision based on concrete information. To determine which lab was better run, they had to significantly improve the way they looked across multiple program labs to compare operating costs, performance and other key metrics. Their contractors also were unable to provide the needed metrics to compare operations—impossible, they said, because the labs used different technologies and because they tested different equipment and had completely different workloads.

Such conclusions need to be revisited, especially given the importance of software to our weapon programs and soldiers. You wouldn't tell an automotive manufacturer that it can't compare two factories because one builds compact cars and the other builds SUVs. In fact, determining the most meaningful metrics for decision makers in the software integration labs will come by examining operations with similar processes, such as the aforementioned automotive factories. These factories input parts, assemble them, and output completed vehicles. The labs input software code and hardware, run tests against the code, and put out a report on whether the code is good or bad. The processes are similar, and the metrics can be similar as well (see Figure 2).

These metrics—capacity, efficiency, effectiveness, and capability, derived from the body of work in manufacturing excellence—will enable decision makers not only to measure and improve each software lab's cost and performance but to manage all their labs effectively as they test the software systems that are fast becoming the strategic weapons on which the military's future success depends. Although these metrics are not yet completely adopted by decision makers who manage software integration labs, they are used throughout automotive manufacturing and are recognized as paramount by executives running similar operations across industries.

As manufacturing improved, a discipline known as overall equipment effectiveness (OEE) was developed to measure how effectively a process was executed. The metrics were designed to allow leaders to compare processes across factories and industries and to provide metrics that decision makers needed to understand if they were to manage their

Figure 1. The Amount of Software in Military Avionics Systems Has Skyrocketed



Note: Source lines of code for the F-16 and F-22 are at first operational flight. F-35 source-line data are from first test flight and from current estimates and sources.
Sources: "Delivering Military Software Affordably," *Defense, Acquisition, Technology, and Logistics*, March–April 2013; A. T. Kearney analysis.

businesses and operations. The meaningful metrics for integration labs closely follow the OEE framework, with tweaks to make them more relevant for software and hardware development.

It is easy to see why these metrics are equally appropriate for measuring lab operations. The comparisons are straightforward. For capacity, auto manufacturers look at the number of cars produced per hour; labs look at the number of test points executed per hour. For efficiency, manufacturers check the number of "lemons" produced per hour; labs check the number of tests executed "on condition." For effectiveness, manufacturers count the number of quality assurance fixes; labs count the number of software defects. For capability, manufacturers explore the functionality of their equipment and what each lab can make; labs explore the abilities of each lab to meet the overall program requirements.

Figure 2. The Metrics for Manufacturing and for Software Testing Labs Are Similar

	Production Metrics	
	Manufacturing	Software Integration Labs
		
Capacity	Number of cars produced per hour	Number of test points executed per hour
Efficiency	Number of good cars produced per hour	Number of tests executed on condition
Effectiveness	Number of quality fixes	Number of defects found
Capability	What can the factory produce? (for example, Porsche vs. Yugo)	What areas and complexity of tests can the lab execute?

These metrics can give program leaders the kind of manufacturing-environment benefits that are valuable in software integration lab measurement, including:

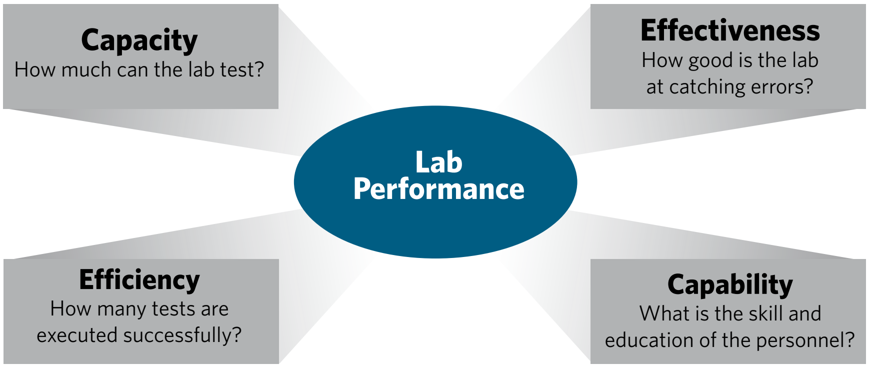
- **Transparency**. With a clear, communicable set of metrics, program leaders can quickly and accurately assess performance and capacity. In addition, fact-based, apples-to-apples comparisons will enable them to contrast each lab's performance against that of other labs.
- **Cost savings**. Cost advantages between labs, which have historically been buried beneath nonrelevant metrics, will be clear when decision makers use equal, meaningful metrics that highlight cost-saving opportunities within the current environment.
- **Risk mitigation**. The metrics will take into account current and future lab capacity, allowing for more accurate estimates of cost and potential schedule delays.
- **Negotiations support**. The metrics will provide the facts on which the best negotiations are based and enable the program office to accurately size and negotiate requirements for contracting labs.

Following is a look at the four main metrics (see Figure 3).

Measured in test points, capacity is the software lab's throughput per hour in terms of its ability to execute its raw work, which includes integration, verification and registration tests. If the lab runs 24 hours a day, seven days a week, how much work could it get done in total units?

Test points can easily be converted into derivative metrics, such as shift capacity, daily capacity and yearly capacity. As the best proxy for lab size, capacity shows whether the lab corresponds to a big or small factory. Knowing a lab's capacity will, among other things, enable planners who are considering shifting work between labs to decide whether

Figure 3. Meaningful Metrics for Software Testing Labs Should Follow an OEE Framework



Note: OEE is overall equipment effectiveness.
Source: A. T. Kearney analysis.

the receiving lab has the maximum capacity to handle the additional work.

Because test points are the basic unit of lab production, comparing dollars per test point is the core indicator of cost in a lab. Using this comparison, decision makers can determine, for example, how much it costs to run a test or how much it costs to find a defect—whether the defect is major (could ground an aircraft) or minor (could prevent a vehicle's windshield wipers from working).

Efficiency is a quality metric that indicates how well the lab is doing the work. If the lab can do 100 units of work in a day but, on average, only 50 come out correct, then the lab's efficiency metric is quite low.

This measurement of the lab's testing procedure shows how many tests must be run before the lab starts finding errors in the testing procedure. The accuracy of this metric depends on several issues, including the quality of the code being input into the lab.

Capability is the skill set of a lab's workforce and the functionality of its equipment. Capability is used to compare how well each lab can test specific areas of the software and is the result of three factors:

- **Knowledge** is assessed across product, functions, and technology, and is proven through work experience requiring expertise in the product, function and technology areas.
- **Competency** is assessed across current work behaviors and skills required to perform the work and proven by the existence of artifacts, such as current job descriptions and training, which are used to validate managers' and directors' scores for their teams and specific knowledge areas.
- **Capacity** is measured by the availability and readiness of the lab's resources (human and infrastructure) to perform an activity.

A business case analysis such as the one done for the DoD can capture a series of deliverables that help leaders better manage their labs and make cost-saving changes that do not hinder the capabilities.

Efficiency is measured with the on-condition metric. "On-condition" is defined as a test executed successfully, according to the checklist and setup procedures handed down by the system engineers, that does not need to be repeated. Efficiency measures the percentage of tests executed correctly—not whether the software being tested passed or failed the test—and is calculated by dividing test points on condition by total test points attempted. "Off-condition" is defined as a test that must be performed again because of an error in testing methods or setup. A false on-condition test is properly executed on condition, but further analysis shows the test package was poorly designed, so the test must be repeated.

Lab capacity and efficiency are tightly linked and are often measured together to provide a clear understanding of their combined effect. Baselines derived from this combination enable leaders to begin making command-level decisions about questions such as how a given action would change the lab's throughput, how a different action would affect the lab's cost per hour or cost per defect and how yet another action would impact the lab's efficiency or capacity.

Effectiveness points out how good the lab is at discovering errors. If an integration lab's primary purpose is to find defects or certify code, the ratio of work units to defects could be a measure of effectiveness. Effectiveness is measured by the number of test points executed per defect found and is calculated by defect found divided by test points attempted.

Because capability is also directly affected by the lab's equipment composition, this composition must be analyzed in any lab-to-lab comparison.

Capability plays a major role in the program leaders' overall management decisions because it has an implicit effect on the other three meaningful metrics. Therefore, its impact on each of these metrics must be understood before making changes to the size, experience or skill set of the workforce.

Meaningful Metrics for DoD

These meaningful metrics for software integration labs were recently used for a DoD laboratory that tests large, complicated systems. The lab had a complex software- and system-testing environment that lacked performance transparency.

The meaningful metrics were developed during the assessment to enable appropriate comparisons across the current lab footprint, which spanned multiple sites with differing approaches to software integration testing. They provided the necessary method to accurately measure and compare lab

performance across the footprint and were essential to the DoD.

In essence, the metrics drove the study, allowing the direct lab comparisons needed for the analysis. With them, the team created a business case to model future scenarios and compare cost savings, transition risks, and steady-state capacity risks across scenarios.

Approach

The assessment objective was to evaluate the current strategy for software integration labs and explore alternative models that might deliver better value. Specifically, the assessment was designed to reduce the life-cycle costs of the labs by moving testing from its current lab to potential alternatives and to do so without degrading current performance.

It also was designed to answer four key questions:

- What are the key attributes of the current lab footprint?
- What are the proposed alternatives to the current lab environment?
- What are the costs, benefits and risks of the current plan and the proposed alternatives?
- What is the recommended strategy (current plan versus proposed alternatives)?

The objective was met with a thorough analytical review of the current long-term strategy and potential alternatives and was shaped by qualitative insights gained during the assessment. The best value alternative would result in the lowest life-cycle cost with manageable risk while not degrading lab capabilities or performance.

Results

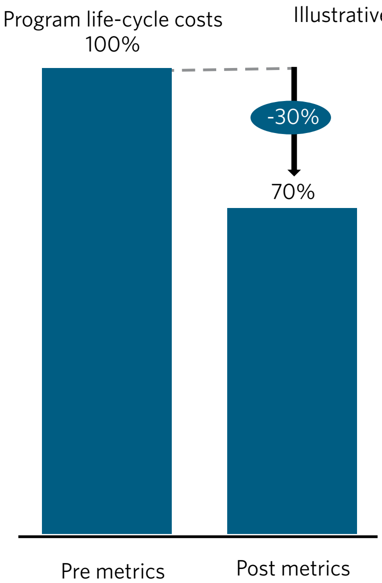
The team recommended that the DoD transition testing from its current lab to alternative labs while maintaining the same performance and the same operator and equipment capability as the current lab, resulting in less risk during transition and normal operation.

The recommendation would also reduce program life-cycle costs by more than 30 percent, for a total net present value savings of hundreds of millions of dollars (see Figure 4).

The team also created clear, communicable metrics that would reflect lab capacity, efficiency, effectiveness and capability—and allow leadership to manage the labs more effectively.

Finally, the team modeled various courses of action from the present day through the perceived end of life. And it recommended a clear course of action for the transition, including the expected cost savings, transition risks and operational risks.

Figure 4. Focusing on Meaningful Metrics Can Reduce Life-Cycle Costs



- Using meaningful metrics, an avionics defense program identified lower-cost labs to perform work
- By shifting work to these locations, life-cycle costs dropped 30 percent
- The lower-cost labs are not always driven by lower labor costs but are evaluated by total cost to test, which includes:
  - — Process
  - — Efficiency
  - — Lab and test philosophy
  - — Equipment requirements

Source: A.T. Kearney analysis

Where Can This Help?

A business case analysis such as the one done for the DoD can capture a series of deliverables that help leaders better manage their labs and make cost-saving changes that do not hinder the capabilities. Potential deliverables include:

- **As-is baseline**: an evaluation of the current-state capacity, efficiency, effectiveness and capabilities of the software integration labs and the development of relevant metrics that will allow further insights into the lab footprint
- **Cost-saving opportunities**: analytical comparisons between labs revolving around the proven metrics, the ability to quantify and estimate previously hidden proficiencies and the generation of plausible future-state scenarios
- **Scenario modeling**: analytical modeling for each of the potential variables; sensitivity, tipping point and worst-case analysis around key input variables; and risks to schedule and the estimated costs to mitigate schedule delays

Software labs, which are expensive and vital DoD assets, often suffer from testing overruns, under-deliveries on initiatives, and intricate projects that make software testing and laboratory management complex. Meaningful metrics can reduce or resolve such problems. These metrics are relevant to a number of different applications faced by software and lab program managers, who might want to consider refreshing their lab performance metrics to realize several objectives in the areas of lab performance, transparency, monitoring and continuous improvement.

Additionally, these meaningful metrics will give DoD technology leaders the information needed to develop a baseline of their current operations, with which they can put in context their decisions and the impact of those decisions. With this baseline, they will know the effect of making small changes, such as how adding capacity will affect a lab's costs, how reducing costs will change the lab's capability and efficiency, and how hiring employees with different skill sets will change the on-condition efficiency. They will know the effect of command-level decisions, such as those they must make when answering questions about whether the labs are effective, whether they have talent or skill deficiencies, or whether significant changes need to be made to improve overall software testing. And, what is perhaps most important, they will know whether their throughput and quality meet the demands of the DoD and individual defense programs.

Finally, these metrics will cut through the confusion that leaders now feel and give them the concrete measures they need for making decisions, not just on technical performance and operations but on fiscal performance. As the DoD's capabilities in developing software and integrated systems mature, these metrics will become even more vital in the department's overall effort to drive efficiencies and savings in its programs to give the warfighter the best, most advanced systems available anywhere.

The authors can be reached at **christian.hagen@atkearney.com**, **steven.hurt@atkearney.com** *and* **andrew.williams@atkearney.com**.

Buying What Works

Case Studies in Innovative Contracting Released

The first version of *Innovative Contracting Case Studies* was released Aug. 21 by the White House Office of Science Technology Policy (OSTP) and the Office of Management and Budget's Office of Federal Procurement Policy (OFPP). "*Innovative Contracting Case Studies* is an iterative, evolving document that describes a number of ways federal agencies get more innovation per taxpayer dollar under existing laws and regulations," according to a joint OSTP-OFPP announcement.

"For example, NASA has used milestone-based payments to promote private sector competition for the next generation of astronaut transportation services and moon exploration robots." the announcement stated. "The Department of Veterans Affairs issued an invitation for short concept papers that lowered barriers for nontraditional government contractors, which led to discovery of powerful new technologies in mobile health and trauma care. The Department of Defense has used head-to-head competitions in realistic environments to identify new robot and vehicle designs that will protect soldiers on the battlefield."

Over the years, there has been much progress on helping federal agencies gain greater access to the innovation and synergies generated by the commercial marketplace. Still, the standard procurement processes on which agencies rely to meet most of their needs may remain highly complex and enigmatic for companies that are not traditional government contractors. Many such companies can offer federal agencies valuable new ways of solving longstanding problems and cost-effective alternatives for meeting everyday needs.

As budgetary constraints continue to reduce available resources, the need increases for new innovative contracting models that can help agencies reach these entrepreneurs and reduce the complexity and cost of doing business with the government. "Such tools allow federal agencies to pay contractors for results, not just best efforts," the announcement stated.

The document stated that the OSTP and OFPP "seek to encourage greater innovation in federal contracting. ... OSTP compiled the collection of agency case studies to highlight different models that have been successfully tested by agencies to meet a range of needs related to research, prototyping, and market testing."

In the joint announcement, officials of OSTP and OFPP said: "We encourage both private sector stakeholders and public servants to engage in a sustained public discussion, identifying new case studies and improving this document's usefulness in future iterations. At the same time, federal government employees can join a community of practice around innovative contracting by signing up for the new 'Buyers Club' e-mail group (open to all .gov and .mil e-mail addresses). This 'Buyers Club' group should provide a useful forum for troubleshooting and sharing best practices across the federal government, serving everyone from contracting officers with deep expertise in the Federal Acquisition Regulation (FAR) to program managers looking for new ways to achieve their agencies' missions."

Note that OSTP compiled these case studies based partly on feedback from external experts, and that the *Innovative Contracting Case Studies* document does not necessarily reflect the views of the federal departments and agencies that are cited as examples. The availability and use of different innovative contracting methods will require consideration of legal authorities and the desired outcome/goals of the specific activity, the study cautioned.

See:

- http://www.whitehouse.gov/blog/2014/08/21/buying-what-works-case-studies-innovative-contracting-0
- Summaries: Find summaries of programs collected at the following URL:
  - http://www.whitehouse.gov/sites/default/files/microsites/ostp/innovative_contracting_case_studies_2014_-_august.pdf

Avoiding Proprietary Problems

A Software Clean-Room Method

Don O'Neill

Heads up! With 80 percent of government software procured as commercial off-the-shelf (COTS) and accorded limited or restricted rights, government acquisition managers need to be alert to intellectual property considerations. When modified and extended through government funding, COTS software becomes government off-the-shelf (GOTS) software entitled to government purpose rights. Unless the government acquisition manager insists on it, a contractor may engage in false claims practice by improperly marketing and selling GOTS software products as COTS. So instead of receiving the benefits of government purpose rights, the government may be charged a commercial product licensing fee and accorded only limited or restricted rights. Neglecting intellectual property rights can be costly!

O'Neill *was president of the Center for National Software Studies (CNSS) from 2005 to 2008. Following 27 years with IBM's Federal Systems Division (FSD), he completed a three-year residency at Carnegie Mellon University's Software Engineering Institute (SEI) under IBM's Technical Academic Career Program and has served as an SEI Visiting Scientist.*

Clean-Room Software-Engineering Summary

Proprietary information may taint a software product. This can occur when commercial off-the-shelf (COTS) software for which a commercial fee is paid for each use is modified or extended through government funding and becomes government off-the-shelf (GOTS) software entitled to government purpose rights following a one-time commercial fee. The difficulty arises when the contractor engages in false-claims practice by improperly marketing and selling GOTS software products as COTS, charging a repetitive commercial fee for each use.

"Clean-room" involves copying a design by reverse engineering, followed by redeveloping the code without infringing on the copyrights and trade secrets present in the original. In an effort to return the software to a permissible fee-bearing commercial off-the-shelf (COTS) status, a vendor may choose to develop a clean-room version free of proprietary information; hence, the need for a rigorously defined clean-room method to transform a proprietary-laden dirty system into a provably correct proprietary-free clean system, one convincingly devoid of reliance on proprietary information, copyrighted material and trade secrets—and not considered a derived work.

Clean-room software engineering entails the reengineering of the dirty system beginning with the production of a proprietary-free hand-over specification and its review by a lawyer to assure no proprietary information, copyrighted material or trade secrets are included or relied upon. The clean-room software-engineering team then prepares proprietary-free artifacts associated with functional specification, usage specification, increment planning, correctness verification, usage modeling, test planning, statistical testing and certification. The kernel of clean-room software-engineering assurance is trusted software engineering using structured programming with its rigorous and provably correct use of zero-and-one predicate prime programs along with proper programs composed of multiple prime programs limited to single entry and single exit.

Framing the Issue

Government acquisition managers sometimes overlook intellectual property considerations. GOTS products are often the result of COTS products extended, expanded and upgraded under government funding to operate in new and changing environments. The GOTS products may even be entered into a company's parts number database. As a result, these GOTS products may contain proprietary information, copyrighted material and trade secrets that may serve to limit or restrict the future use of the GOTS products. In effect, proprietary information, not unlike malware, may taint a software product.

Why is that a problem? COTS products represent more than 80 percent of government software procured. The originating commercial organization may attempt to restrict use of the proprietary-based COTS product and even an enhanced GOTS product in order to assert a competitive advantage in a downstream procurement with the objective and intended outcome of locking in a sole-source contract. Two software assurance challenges present themselves. The first challenge is how to detect proprietary information, copyrighted material or trade secrets. The second challenge is how to convincingly assure the clean provenance of GOTS products in such an environment.

Government systems and components may contain proprietary information, copyrighted material or trade secrets that serve to limit or impede use or reuse. A contractor may unintentionally drift into using such tactics only to find that it can exploit and leverage its proprietary information in later procurements and then attempt to do so. Beyond that, and perhaps less likely, a contractor may intentionally and stealthily seed proprietary information in systems and components only to reveal the proprietary presence later for the purpose of locking in its solution for future work. Government-funded systems and components also yield government-owned proprietary systems and components that contractors may attempt to package and resell as their proprietary products on the global market. Upon detecting a dirty system, a contractor may choose to shed the restriction by redeveloping the dirty GOTS software into a clean system. This can be done by employing a rigorously defined "clean-room" method to transform a proprietary-laden dirty system into a proprietary-free clean system, one devoid of reliance on proprietary information, copyrighted material, or trade secrets and not considered a derived work. Such a method employs both a "Chinese wall" protocol of separation and the clean-room software engineering technology and process.

To avoid false claims charges, a contractor is advised to segregate core commercial software components produced by private funding from those produced through government funding and to do so at the lowest practical level.

There are many questions that a government acquisition organization needs to ask and answer to begin focusing on its intellectual property practices. To what extent are COTS products enhanced with government funds resulting in GOTS products? When COTS products are enhanced with government funds, resulting in GOTS products, is it the contractor's practice to enter the GOTS product into the parts number database or to append its own copyright? To what extent do such practices go undetected? To what extent are systems and components trusted with respect to contractor proprietary information, copyrighted information or trade secrets that could limit or impede use or reuse? To what extent are systems and components trusted with respect to government-funded proprietary information that could result in unauthorized release of GOTS

as contractor-proprietary products? To what extent do systems or components thought to be GOTS have restrictions on downstream use or reuse? To what extent should proprietary information concerns be included in the approach to trusted systems and networks and their supply chain risk-management (SCRM) assurance? To what extent should the National Institute of Standards and Technology Special Publication 800-161 SCRM Plan address the identification and risk mitigation of government systems and components that may contain proprietary information, copyrighted material or trade secrets that limit or impede use or reuse, lock in contractor solutions for future work or facilitate the packaging and reselling of GOTS as contractor-proprietary products?

False Claims Violations

An organization may cross the line and be guilty of false claims violations when it markets, sells, deploys or delivers a version of its product produced by mixed funding, company investment and government funding as commercial software with limited or restricted rights—thereby depriving the government agency of the government purpose rights it may have purchased and deserved. In an environment of ascending demand for a software product, a company may commit numerous false claims violations during rollout when it improperly markets, sells, deploys or delivers such a software product as a commercial product with limited or restricted rights rather than properly as a noncommercial software product with government purpose rights.

Faced with a mixed funding history in a software product, a company may elect to produce a commercial version by reengineering the software product in question using the clean-room method and process. Once a clean-room project has been undertaken, the object of marketing and selling shifts to the clean-room version of the product under development with a future delivery date, not subject to charges of false claims violations. In effect, the company is insulated from charges of false claims violations during the window of "under development." Since clean-room reengineering is challenging and difficult, the clean-room project schedule may slip and become extended, exceeding the original estimate and plan, thereby setting up a dilemma for the company facing firm delivery commitment deadlines to customers performing in mission-critical operations. A company that has made a commitment to deliver a commercial product with limited and restricted rights and that is faced with an incomplete clean room may decide it has no alternative but to deliver the dirty system, the product of mixed funding, and may do so without reverting to government purpose rights—a false claims violation. Of course, it takes two to Tango and such an outcome must involve the company's intent and the government acquisition contract officer's and program manager's neglect of due diligence in accepting and relinquishing data rights asserted by the contractor.

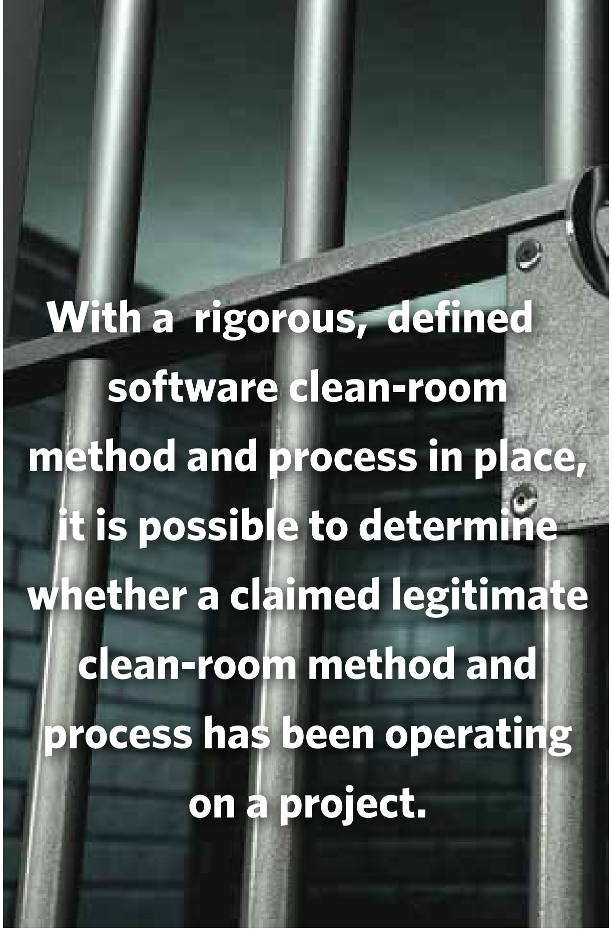
In order for the clean-room window "under development" to insulate the company from numerous charges of false claims violations, the clean-room method and process must be bona fide and legitimate—that is, the clean-room method and process must ensure an environment and operation devoid of any use or knowledge of proprietary means or methods used in a predecessor implementation.

The Need for a Clean-Room Method and Process

Rigorously defined clean-room method and process are needed to transform a proprietary-laden dirty system into a provably correct proprietary-free clean system—one convincingly devoid of reliance on proprietary information, copyrighted material or trade secrets and not considered a derived work; one with methods of investigating legitimacy, confirming intent and wherewithal of people, verifying process execution and validating outcomes in determining that a legitimate clean room was in operation—nd one with an outcome based on trusted software engineering principles and practices in producing provably correct software components.

Goal

The clean-room method and process are intended to assure an environment and operation devoid of any use or knowledge of proprietary means or methods used in a predecessor implementation. A Chinese wall is used—and management, specification, development and certification personnel are involved. Clean-room method and process assurance encompass an explicit statement of intent and adherence to specified practices. These include intellectual property practices, protocol of separation, clean hand-over specification process, clean-room software engineering process and software clean-room investigation process. These practices form the basis for the



assurance, assessment, examination and investigation of an organization's clean-room method and process spanning confirmation through people, process execution-based verification and outcome-based validation.

The essential focus of a software clean-room investigation revolves around the following questions:

- Was there clean project access to the target code comprising the direct expression of the copyright material, and was there substantial similarity exhibited by the clean system?
- Does the clean system stand up to the abstraction-filtration-comparison (AFC) test, a three-step process for determining substantial similarity of the nonliteral elements of a computer program?

Specification

Defined software clean-room methods and processes are needed to transform a software system or component based on proprietary information, copyrighted material or trade secrets to a functionally equivalent clean software system or component devoid of any traces or reliance on the proprietary information, copyrighted material or trade secrets. The proprietary system is termed the dirty project and the transformed system is termed the clean project. The challenge is to insulate the clean system from the dirty system so it will not be considered a derived work in form or function. What are the criteria for a legitimate clean room? A rigorous, defined software clean-room method employs both a Chinese wall and the clean-room software-engineering technology and process.

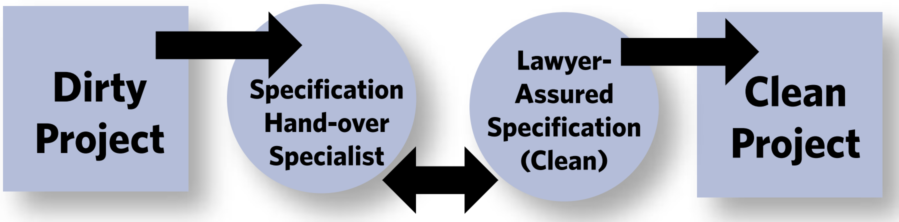
Defined Clean-Room Method

The method features a multidimensional Chinese wall spanning defined protocols of separation for physical location, people, electronic infrastructure and software development tools. The Chinese wall is composed of a clean environment demonstrably uncontaminated by any proprietary information or knowledge of such through a discipline of multidimensional separation.

The clean environment begins with physical separation in a separate, undisclosed location. Separation extends to the personnel participating in both the dirty project and the clean project, including their training and organizational policy governing the rules of separation and no contact. Separation extends to the electronic infrastructure including e-mail and telephone access and online document access and to the software development tools employed, including text editors, programming language, language compilers, test suites, configuration management tools and parts number database management tools.

The clean-room software-engineering process extends the discipline of multidimensional separation through discrete teams for management, specification, development and certification of the clean system. Clean-room software-engineering entails the reengineering of the dirty system beginning with the production of a proprietary-free hand-over specification and its review by a lawyer to assure no proprietary information, copyrighted material or trade secrets are included or relied upon (see Figure 1). Clean-room software engineering teams prepare proprietary-free artifacts associated with functional specification, usage specification, increment planning, correctness verification, usage modeling, test planning, statistical testing and certification. The kernel of clean-room software-engineering assurance is trusted software engineering using structured programming with its rigorous and provably correct use of zero-and-one predicate prime programs along with proper programs composed of multiple prime programs limited to single entry and single exit.

Figure 1. Specification Hand-Over Procedure Involving Hand-Over Specialist and Assurance Lawyer



Outcome-Based Validation

Improper use of proprietary software involving proprietary information, copyrighted material and trade secrets increasingly goes undetected. Uncovering such use and detecting specific instances of substantial similarity is a technical challenge that usually requires full and ready access to the dirty system source code for best results as well as the wherewithal and means to express the proprietary information, copyrighted material and trade secrets in a precise, rigorous and trusted abstract manner suitable for computer searching and comparison.

Proprietary software is licensed under exclusive legal right of the copyright holder with the intent that the licensee is given the right to use the software only under certain conditions and restricted from other uses. In the legal community, the AFC test is a three-step process for determining substantial similarity of the nonliteral elements of a computer program. Abstraction distinguishes which aspects of the program constitute its expression, and which are the ideas. Filtration removes from consideration aspects of the program that are not legally protectable by copyright—such as elements associated with efficiency, external factors and the public domain. Comparison considers whether just those aspects of the program that constitute its expression and not those aspects not legally protected by copyright are present in the clean system.

In addition, proprietary information, copyrighted material, and trade secret detection can potentially be determined using NIST's approximate matching text strings to detect

Table 1. Software Clean-Room Investigation Process: Confirmation

Confirmation	Software Clean-Room Investigation Process
Intellectual Property Practices	1. Had the original commercial off-the shelf (COTS) product been entered into the parts number database earlier? Was it assigned a unique number? What was that number? 2. Had the dirty system been entered into the parts number database earlier? Was it assigned a unique number? What was that number? 3. Was the clean system entered into the parts number database? Was it assigned a unique number? What was that number? 4. Was a copyright notice appended to the original COTS product? What was the copyright notice? 5. Was a copyright notice appended to the dirty system product? What was the copyright notice? 6. Was a copyright notice appended to any open source software used? What was the copyright notice?
Statement of Intent	1. Did the project have a clean-room process definition? 2. Was there an explicit management commitment to follow the defined process? 3. In actual practice, was the defined process followed? 4. Did the clean-room process definition include protocols of separation, clean-room software-engineering process, clean hand-over specification process? 5. Was the result a clean system?
Protocol of Separation	1. Was a protocol of separation defined? 2. Was there an explicit management commitment to follow the defined protocol of separation? 3. In actual practice, was the defined protocol of separation followed? 4. Did the defined protocol of separation include physical location, people, electronic infrastructure and development tools?
Clean Hand-Over Specification Process	1. Was there a defined clean hand-over specification process? 2. Was there an explicit management commitment to follow the defined clean hand-over specification process? 3. In actual practice, was the defined clean hand-over specification process followed? 4. Did the clean hand-over specification process include having a lawyer review the clean hand-over specification to assure that no proprietary information, copyrighted material or trade secrets were included? 5. Did any clean system personnel ever have any access to any dirty system code?
Clean-Room Software-Engineering Process	1. Was the clean-room software-engineering process defined? 2. Was there an explicit management commitment to follow the defined clean-room software-engineering process? 3. In actual practice, was the defined clean-room software-engineering process followed? 4. Did the defined clean-room software-engineering process include functional specification, usage specification, increment planning, correctness verification, usage modeling, test planning, statistical testing and certification?

fragments. More advanced, Carnegie Mellon University's function extraction for abstracting intended function and Oak Ridge National Laboratory's Hypernion using behavior specification units (BSUs) for detecting intended function offer promise in this space. The Defense Advanced Research Projects Agency's Mining and Understanding Software Enclaves (MUSE) program incorporates a continuously operating specification-mining engine to conduct deep program analysis on the corpus of software drawn from the hundreds of billions of lines of open-source code to identify and understand deep commonalities.

Operations and Risks

With a rigorous, defined software clean-room method and process in place, it is possible to determine whether a claimed legitimate clean-room method and process has been operating on a project. Numerous clean-room method and process risks must be assessed. Does the organization explicit commitment match the intent and means employed? Do the means employed match the defined software clean-room method and process? Does the protocol of separation ensure verifiable separation under all circumstances of use? Do the actual organization intellectual property practices reveal the organization's intellectual property ownership intentions? Is the clean hand-over specification process with lawyer-assured clean specification confirmed through people and verified through process execution evidence? Is the clean-room software-engineering process verified through process execution evidence? Are clean-room method and process execution outcomes validated through clean product results achieved devoid of proprietary information? Was there clean project access to the target code comprising the direct expression of the copyright material? Was there substantial similarity to the target code exhibited by the clean system?

Conclusion

With the rigorous, defined software clean-room method and process specified, the question of whether a legitimate clean room was in place and operating can be addressed by confirming the equivalency of the intent and means employed, verifying the extent to which the defined protocols of separation were practiced, validating the clean-room software-engineering process execution and outcome with respect to convincingly achieving the intended result of a proprietary-free clean system, and reporting the results in terms of findings, rationale, recommendations and consequences.

Confirmation that a software clean-room investigation process was undertaken begins with obtaining answers to the pertinent questions (see Table 1). Other more probing questions focus on verification through process execution and validation through outcomes achieved.

The author can be contacted at **oneilldon@aol.com.**

SECTION 3685, TITLE 39, U.S.C. SHOWING OWNERSHIP, MANAGEMENT, AND CIRCULATION

Defense AT&L is published bimonthly at the Defense Acquisition University, Fort Belvoir, Va. 22060-5565. The university publishes six issues annually. The director of the DAU Press is Randy Weekes; the managing editor of *Defense AT&L* is Benjamin Tyree; and the publisher is the Defense Acquisition University Press. All are colocated at the following address: Defense Acquisition University, Attn: DAU Press, 9820 Belvoir Rd., Ste. 3, Fort Belvoir, VA 22060-5565.

Average Number of Copies Each Issue During The Preceding 12 Months

A. Total number of copies printed (net press run):	3621
B. Paid and/or requested circulation:	3513
1. Sales through dealers and carriers, street vendors, and counter sales:	0
2. Mail subscriptions paid and/or requested:	3513
C. Total paid and/or requested circulation:	3513
D. Free distribution by mail, carrier, or other means; samples, complimentary, and other free copies:	75
E. Total distribution:	3588
F. Copies not distributed:	33
G. Total distribution:	3621

Actual Number Copies of Single Issue Published Nearest To Filing Date

A. Total number of copies printed (net press run):	3607
B. Paid and/or requested circulation:	
1. Sales through dealers and carriers, street vendors, and counter sales:	0
2. Mail subscriptions paid and/or requested:	3491
C. Total paid and/or requested circulation:	3491
D. Free distribution by mail, carrier, or other means; samples, complimentary, and other free copies:	70
E. Total distribution:	3561
F. Copies not distributed:	46
G. Total distribution:	3607

General TCF Closure Tasks

in the U.S. Army Signal Corps

Capt. Jeffrey P. Stevens, USA

The 198th Expeditionary Signal Battalion (ESB) provided unparalleled communications support to the warfighters during its 2013–2014 deployment to Afghanistan. The ESB provided tactical satellite communications, network operations expertise, and cable and wire services. This National Guard Battalion is comprised of three units from Delaware and a fourth from South Carolina. The Battalion faced the unique challenge of learning how to close a Technical Control Facility (TCF). The Battalion met this daunting task with detailed preparation and coordination, effectively closing four TCFs.

Stevens *was the assistant operations officer for the 198th Expeditionary Signal Battalion (Delaware Army National Guard) under Task Force Signal, 160th Signal Brigade in Kandahar, Afghanistan. He is an Army civilian systems engineer employed by the Army Software Engineering Center—Communications Electronics Command (SEC CECOM).*

Left: The interior of a Technical Control Facility (TCF), which provides local services such as e-mail, sharepoint, telephone routing and file storage.

Below: A mini-TCF previously located at Forward Operating Base Pasab in Afghanistan is shown divided in half in order to be shipped off site. The missing section is the mirror image of the right-most displayed section. A TCF supports NIPR, SIPR and CX-I nonsecure and secure Internet and combined information-exchange services.
Photos by the author

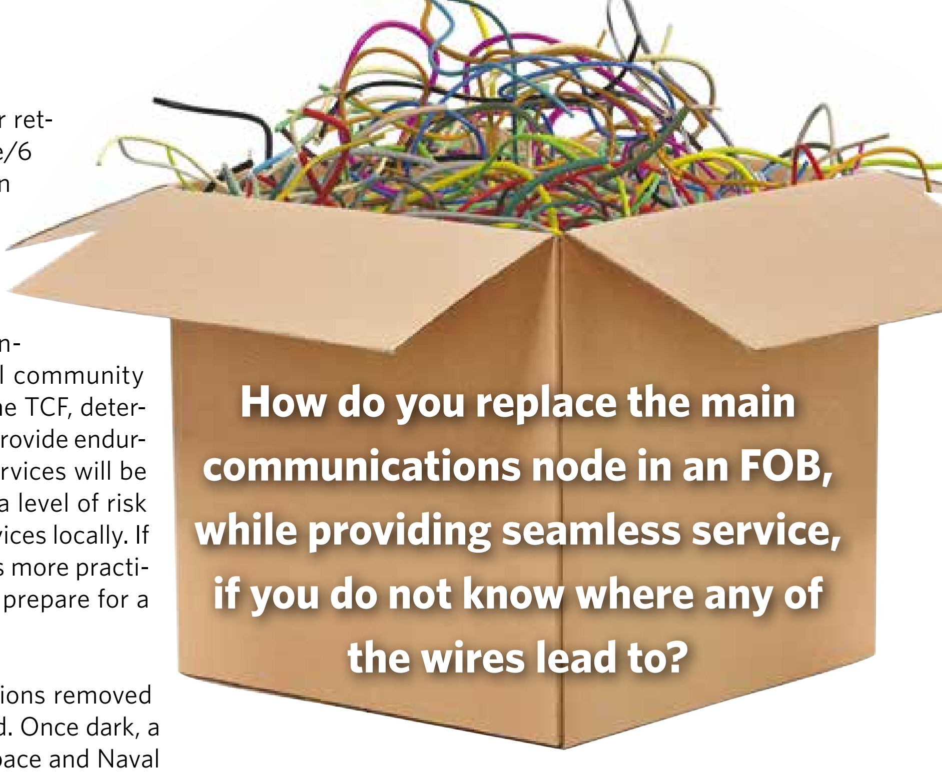
A TCF provides network services to large user bases in tactical and strategic environments. E-mail, file storage, phone routing, host-based security system (HBSS), active directory (AD) and domain name system (DNS) are key services delivered to users while units are deployed tactically. A TCF can be fixed or modular and miniature to medium size. A miniature TCF can service up to 4,000 customers while a medium-size TCF can service up to 20,000. A typical TCF allows customers to access non-secure internet protocol router network (NIPR), secure internet protocol router network (SIPR) and the combined enterprise regional information exchange system—Afghanistan (CX-I). During this deployment, the 198th ESB retrograded four modular TCFs—three miniatures and one medium. The TCFs were packed up and shipped to other locations via ground and air movement. The services they provided were replaced with a customized tactical solution that encompassed a smaller footprint.

During the initial preparation of a closure, it is imperative that the Battle Space Owner is intricately aware of all facets of the plan and the expected impact on warfighter communications. The challenge is to ensure that the user's services are not interrupted during their migration to either local or regional hub sites, such as Kandahar Air Field in Afghanistan. To accomplish this objective, a temporary set of computer servers and file servers must be created from scratch with theater-provided equipment. The data stack is configured to each site's specific needs and deployed at the tactical out site

where the TCF in question was identified for retrograde. Redundant fiber and category 5/5e/6 network cable must be run from every location at the Forward Operating Base (FOB) to the new data stacks, all while ensuring the TCF network remains intact.

Once redundant or backup services and connections are established on site, the Signal community within the Regional Command comprising the TCF, determines if the installed custom data stack will provide enduring services, or if a portion or all of those services will be fully migrated to a major hub site. There is a level of risk associated with not terminating network services locally. If the FOB is nearing complete closure, then it's more practical to migrate services to a distant hub and prepare for a complete closure at that location.

The TCF goes dark and all network connections removed when all site services are properly transferred. Once dark, a 198th ESB retrograde team augmented by Space and Naval Warfare Systems Command contractors arrive to dismantle the TCF. Although planning allows for several weeks for these actions, an efficient team, under the right conditions, can dismantle a TCF in four to five days and have the site totally clear. Proper planning with the network migration enables the well-trained retrograde team to work quickly at inventory, teardown and shipping.



FOB Spin Boldak TCF Closure

The FOB Spin Boldak TCF closure presented our team with a unique set of challenges. The FOB was comprised of an unlabeled cable backbone built by multiple units over several years. After years of operation and more than a dozen units stationed in this FOB, the network was a complicated mess. How do you replace the main communications node in an FOB, while providing seamless service if you do not know where any of the wires lead to? The FOB experienced constant fiber breaks due to unmarked cables being dug up and cut, and this resulted in loss of services. Furthermore, improper labeling increased the threat of cross-domain violations (CDVs). The situation was grim and unpredictable.

A cable and wire team was dispatched two months in advance of TCF closure to properly test, label and map the network diagram for the FOB. The four team members worked 12 to 16 hours a day to map and record every single wire going into and out of the TCF. It was painstaking but necessary.

The cable and wire mission consisted of a second critical objective: properly connecting all FOB locations in a logical and commercially modeled manner. Redundant fiber connections were redesigned and new physical network nodes were established throughout the FOB to facilitate a modern star topology. A star topology is a physical network configuration that allows for many redundant links in case of link breakage. It is very important to utilize this type of topology in the tactical environment in order to mitigate any combat-related damages to the network, including those from indirect fire or FOB infiltration via a vehicle-borne improvised explosive device (VBIED). If a line breakage occurs due to this sort of damage, the network would continue to function and the warfighter would continue to communicate during this critical event. More than 40,000 feet of networking cable were run to accomplish this task. The network made sense to the customer and administrator.

Concurrent with the cable and wire mission, a 198th ESB Network Engineering team was creating a data stack for deployment to the FOB. The data stack consisted of all the networking equipment, file storage and computing power required to locally provide file, voice, e-mail and print services on site. It was determined that HBSS, Active Directory and DNS services would be migrated to Kandahar Airfield. The migration of those services to Kandahar would be complete before the data stack was deployed.

After two months of cable and wire migration, and one month of assembling and configuring the custom data stack, the site was prepared to transfer services locally. The data stack was sent out with both a network-engineering (NetEng) and enterprise-operations (EntOps) team. The NetEng team was responsible for connecting the stack to the network and ensuring all connections to customers were complete. The EntOps team set up the services and ensured that the local communications team was properly trained on its operation. The cable and wire team was on standby to repair any connections that may have been overlooked during their two months of preparation.

Within one full week of concurrent operation with the data stacks providing primary services and the TCF providing back-up services, the mission was declared a success and the TCF went dark. Cables were cut between the TCF and the

FOB. The data stack was now the primary communication node for the FOB. A 198th ESB retrograde team arrived to dismantle the TCF within four days. Spin Boldak's TCF Closure was a complete success with no interruption in services to the warfighter.

TCF Closure Lessons Learned

There are a few lessons learned from the 198th ESB's four TCF closures:

A cable and wire team should be dispatched as early as possible with a representative from the NetEng team building the data stack. Collaboration between the cable team and the engineers was crucial in order to develop a logical migration plan. Depending on the state of the fiber network at the FOB, the cable and wire team must be on site anywhere from two weeks to two months. There was a large difference in network maturity and complication between FOBs. No two are alike.

Ensure users are properly informed. Scheduling authorized service interruptions (ASIs) are a key item of which we had to keep FOB and regional Signal Corp leadership informed. It is very important, overall, to develop face-to-face relationships with major FOB customers and Battle Space Owners. In our case, a 198th ESB site officer or noncommissioned officer in charge would personally engage key combatant commanders to inform them of the network status—an essential part of customer service.

Develop a well-rounded team of soldiers with skills in network, movement and heavy equipment operations. In our case, this resulted in total success. Through proper planning and team building, TCF closures can be seamless and painless transitions during a retrograde operation.

The author can be contacted at **jeffrey.p.stevens.civ@mail.mil** *or at* **jeffrey.p.stevens.mil@mail.mil**.

MDAP/MAIS Program Manager Changes

With the assistance of the Office of the Secretary of Defense, *Defense AT&L* magazine publishes the names of incoming and outgoing program managers for major defense acquisition programs (MDAPs) and major automated information system (MAIS) programs. This announcement lists all such changes of leadership for both civilian and military program managers that occurred in recent months.

Defense Information Systems Agency

Russell Daul relieved **Salvatore Scaglione** as program manager for the Department of Defense Teleport program on May 12.

Army

Col. Courtney P. Cote relieved **Col. Timothy R. Baxter** as project manager for the MQ-1C Gray Eagle Unmanned Aircraft System Program on July 11.

Col. Robert M. Collins relieved **Col. Charles A. Wells** as project manager for the Distributed Common Ground System-Army Increment 1 (DCGS-A Inc 1) Program on July 23.

Col. Jong H. Lee relieved **Col. John R. Leaphart** as project manager for the Common Infrared Countermeasure (CIRCM) Program on July 31.

Col. James P. Ross relieved **Col. William R. Wygal** as project manager for the Airborne & Maritime/Fixed Station Joint Tactical Radio System (AMF JTRS) and Joint Tactical Radio System Handheld, Manpack, and Small Form Fit Radios (JTRS HMS) Programs on Aug. 19.

Navy/Marine Corps

Capt. Casey Moton relieved **Capt. John Ailes** as program manager for the Littoral Combat Ship Mission Modules (PMS-420) Program on July 28.

John Karlovich relieved **Robert Bond** as program manager for the Ground Air Task Oriented Radar (G/ATOR) Program on Aug. 1.

Air Force

Col. Kevin D. Hickman relieved **Col. James C. Baird** as program manager for the Small Diameter Bomb (SDB) Program on June 12.

Col. Douglas W. Roth relieved **Col. Brian S. Jonasen** as program manager for the CV-22 Osprey Program on June 13.

Lt. Col. Margaret Barker relieved **Lt. Col. Karl C Schloer** as program manager for the HC/MC-130 Program on June 15.

Col. Stephen G. Purdy relieved **Col. Rodney L. Miller** as program manager for the Advanced Extremely High Frequency (AEHF) Program on June 23.

Col. Peter K. Eide relieved **Col. Dale J. VanDusen** as program manager for the Advanced Pilot Trainer (APT) Program on July 1.

Col. Anthony W. Genatempo relieved **Col. Gregory M. Gutterman** as program manager for the F-22 and F-22 Modernization Increment 3.2B Programs on July 19.

Col. Christopher B. Athearn relieved **Col. William A. Ellis** as program manager for the Joint Air-to-Surface Standoff Missile (JASSM) Program on July 21.

A View From Space
NASA Systems Engineering and Test

Woody Spring

It has been three years since I witnessed the last Space Shuttle launch, STS-135, lifting off from Earth on July 8, 2011. It was the seventh I had witnessed, but this one had special meaning. Twenty-nine years ago, I was on the inside *looking out* as a part of the STS-23 (STS 61-B) crew. I flew Atlantis on her second flight in 1985 and had observed her construction years earlier at Rockwell International's space shuttle-assembly location.

As a crew, we visited the facility in Palmdale, Calif., where the components were finally assembled. It was an awesome spectacle. This was where a reusable, reliable and incredibly powerful rocket ship called Atlantis came alive. Technology was ubiquitous. There were so many critical components that had to be harmonized. If it weren't for systems engineering and its embedded process imperatives though, the shuttle would have never taken off the ground.

In the last six years, in my capacity as a professor at the Defense Acquisition University, I have found myself reflecting more and more about that day and the importance of Systems Engineering and Test as well as the influence NASA has had on Department of Defense (DoD) weapon system developments.

Technical Necessities Influenced Future Technologies

Like many of DoD's weapon systems, every component on the shuttle experienced decades of experimentation and analysis before it found its home on an operational system: the shuttle. Many key materials and processes didn't even exist, but the shuttle would later depend on them to meet the user's requirements. After all, this newly

Spring *is a former astronaut and now is a professor of Engineering and Test for the Defense Acquisition University, West Region, San Diego, Calif.*

combined air and spacecraft had to blast off with an incredible force (40,320 kilometers per hour or 25,000 miles per hour or 7 miles per second) to escape Earth's gravitational pull, easily maneuver in both subsonic and hypersonic speeds, protect its crew in the cold and unforgiving vacuum of space, and return the crew safely to Earth. The shuttle's exterior had to tolerate temperature extremes colder than Antarctica and hotter than the temperature at which most metals melt. Its crew compartment had to protect its inhabitants from the constant bombardment of radiation.

Needless to say, NASA engineers had to push every operating envelope. Over time and decades of component and full-scale testing, the shuttle took shape. It all came together in a unique form. Aerodynamically, it capitalized on the X-24B lifting body from 1975; NASA adopted a similar winged platform configuration with a comparatively low lift-to-drag ratio like the X-24 that could land accurately without power. Like the X-24, a Space orbiter no longer needed an engine after reentry and would become an unconventional glider, given its maximum landing weight of 230,000 pounds.

NASA instituted technical standards that promoted interoperability among its programs. Since each and every experience in space tended to be groundbreaking, NASA captured engineering lessons learned and proven practices. However, in many cases, NASA engineers had to serve as development pioneers, not to mention perpetual problem solvers. Innovation was always a constant priority. NASA engineers had to infuse technology into solutions to keep costs low without trading away capability or personnel safety. Sound familiar in DoD?

Exploiting Technologies

Since the 1960s when President Kennedy first challenged NASA to send a man to the moon and return him safely to Earth, NASA has produced a tremendous array of technical innovations that have given the United States a noticeable and distinctive advantage. Today our country's national defense development community employs many of NASA's technical accomplishments in numerous weapon systems that continue to help our warfighters maintain a distinctive competitive advantage where it matters the most—on the battlefield.

Unlike Teflon, which was accidentally invented by Roy Plunkett of Kinetic Chemicals in 1938 when he tried to make a new refrigerant and the chemicals polymerized in a pressurized storage container, NASA's developments were carefully guided and cultivated. Some of those gains produced by NASA can be seen in:

- Software
  - Critical Path computer software test and evaluation
  - Semiautonomous and fully autonomous systems and control algorithms
- Robotics: Development of artificial muscle systems with robotic sensing and actuation capabilities for use in NASA space robotic and extravehicular activities that have been adapted to create more functionally dynamic artificial limbs
- Aerodynamic control of inherently unstable platforms (the shape, especially of an aircraft, seen from above)
- Hypersonic platforms
- Aviation safety such as onboard diagnostics and integrated sensing/evaluation/warning
- Self-contained exploration sensors
- Management techniques
  - Technology Readiness Levels (TRLs), with linked Software Readiness and Manufacturability Readiness Levels
  - Configuration control processes
  - Program Requirements Management control
  - Modeling and simulation (M&S)
  - Motion-based trainers
  - Joint integrated simulation at multiple sites
- System of systems architectures
- State-of-the art technologies
  - Microprocessors
  - Component miniaturization
  - Biometrics, solar energy
  - Fuel cells
  - Thin film membrane structures
  - Expandable structures
  - Liquid rockets
  - Dynamic rocket and engine control
  - Astrobiology
  - Environmental monitoring
  - Environmental cleanup and sensing
  - Life support

The countless technical advances NASA achieved also found their way into a wide array of commercial products we use every day back on Earth, including state-of-the-art exercise machines, trash compactors, water filters, smoke detectors, solar and tankless water heaters, quartz clocks, bar codes, smaller digital cameras, complementary metal oxide semiconductor chips and technology used in cell phones, cameras, webcams, digital image stabilization, insulating material and other means.

When law enforcement officials needed help improving a grainy crime scene video, NASA assisted with the high-tech image-processing technology it used to analyze space shuttle launch video. NASA also seeded some major industry leaders with game-changing technologies. Goodyear Tire and Rubber Company produced a radial tire with a tread life expected to be 10,000 miles greater than conventional radials by using a fibrous material it developed for NASA.

Process Ruled the Day

As I stood watching the countdown clock for Atlantis, I also remembered the importance of technical and management processes. They ruled the day. Those integrated processes that NASA and DoD share provide a methodology for designing and realizing systems—and for planning,

The author manipulates a structure during the second Extra Vehicular Activity from the Space Shuttle Atlantis.
NASA photo.

assessing and controlling the technical development effort as it evolves.

As astronauts, we practiced every process step along the pathway to ensure all system functions responded to our human actions as intended. Just as it did before, the thousands of coded exchanges that took place between Launch Control Center the day I left Earth in 1985 and the last time the shuttle left Earth on July 8, 2011, affirmed whether every key component could safely "go for launch."

If any component operated outside its performance envelope, "built-in" holds immediately surfaced and delayed the launch until the issue was fully addressed. The tight coupling of technical and management processes that was exercised beforehand reduced the likelihood of lifting off with an unresolved issue.

Diverse Teams Can Overcome Adversity

At high school, West Point, Navy Test Pilot School, my Test Pilot group at Edwards Air Force Base and in Vietnam, I noticed early the significance of teams and the tremendous outcomes they achieve working as a unit. From ground crew to mission crew, the NASA team members were incredibly professional and mission-focused as well as being leading experts in their fields. My astronaut experience reinforced this lesson even more. We learned from our combined knowledge and experience. We benefited from our diversity in much the same way that DoD's acquisition integrated process and product teams do today.

At NASA, we knew we had to depend on each other during our qualification process. We practiced everything over and over until it became second nature. For more than eight years, I had the good fortune to participate in this amazing NASA dynamic that could respond to any technical or leadership challenge, no matter what conditions prevailed. Sad to say, the dangerous nature of space exploration yielded a few tragedies resulting in the loss of wonderfully dedicated and accomplished Americans.

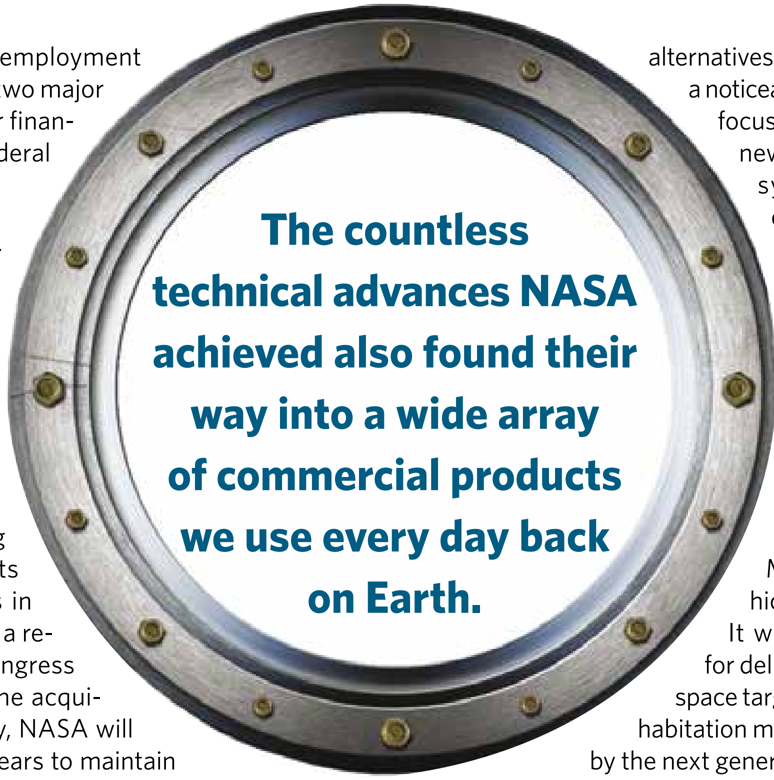
Two shuttle accidents, several aircraft vehicle accidents, and the same medical conditions we all face outside NASA struck some of my NASA colleagues. Every one of them made their mark on history and will forever be remembered by helping make space travel safer and more meaningful.

The Experience Quotient

Technical experience in both DoD and NASA takes time to develop. After the shuttle program formally ended, many of the personnel faced a different kind of fate in the form of impending unemployment. The end of the shuttle era also meant numerous subordinate programs reached the end of their lives as well.

Nevertheless, as the mission at NASA evolved just like it did after the Apollo program ended in 1972, managers worked to place personnel in other jobs and/or explore retraining opportunities. Many of these workers had been supporting the shuttle program since their 20s. Now, with the shuttle program ending, they were in their 50s. Retraining and relocating at this age proved difficult and uncertain for some.

About 30,000 aerospace engineers and support personnel were at risk. The unemployment numbers were an equal concern for other industries across the country. But since 1960 NASA has never seen human capital challenges like those



The countless technical advances NASA achieved also found their way into a wide array of commercial products we use every day back on Earth.

of today. The national unemployment rate, growing deficit, and two major wars have created greater financial pressures for every federal agency, including NASA.

In the mid-1990s, the Defense Acquisition Workforce was cut in half for a number of reasons, including outsourcing. This cutback was expected to create huge efficiencies and savings. There were also many unintended consequences, including serious experience deficits in the government ranks in the following decades. As a result, in 2008 the U.S. Congress passed a law to rebuild the acquisition workforce. Similarly, NASA will be tested in the coming years to maintain its foundation of experience to avoid a similar outcome. Experience matters in every career field, especially systems engineering and test.

The Frontier Forward

When our nation retired the space shuttle, an American icon recognized and envied around the world as the symbol for space over the last quarter-century became history. Is the future of America's leadership in space at risk? NASA faced a similar challenge in the early 1970s when Congress canceled the last three Apollo moon missions with little notice, leading to a major gap in a U.S. launch capability. NASA used one Apollo Saturn V rocket system to build and launch Skylab, but then watched Skylab de-orbit after three missions.

The United States found itself with no launch capability to reboost Skylab to a stable orbit and no gap filler. The shuttle's operational deployment was too late to help. Now, after 30 years of spectacular service, the shuttle is no longer safe to use without a major update of multiple systems. Absent an expensive life-extension program, system reliability was well below acceptable levels for the shuttle.

Like the challenge in the early 1970s, no replacement system is ready to fill the gap in time. With our nation's weapons systems, we had to make equally tough choices but could not afford certain critical operational gaps that would jeopardize warfighting capability. As a result, many of today's weapon systems are in service well beyond their expected life. These include the B-52, which first saw service in 1955.

NASA has decided to get out of the business of Low Earth Orbit (LEO) launch operations because industry and commercial ventures are expected to become more economical alternatives. Space X has already taken a noticeable lead. NASA will instead focus beyond LEO and incubate new technologies. Invariably, systems engineering will continue to predominate.

NASA has two new exciting programs under consideration and tentative development—a Crew Capsule and a heavy lift Space Launch System (SLS). The crew capsule will have a deep space capability with a Multi-Purpose Crew Vehicle (MPCV) and seat four. It will be the primary vehicle for delivering astronauts to deep-space targets. It will also mate with a habitation module and can be launched by the next generation commercial systems or SLS. NASA continues with creative innovation in multiple product lines reinforcing American leadership in next generation technologies similar to what the United States enjoyed following Apollo.

Conclusion

The United States needs to make hard choices if NASA is to send astronauts to an asteroid by 2025, and a crewed Mars mission by the 2030s. Game-changing technologies are still a necessity; and key processes and experience will continue to rule the day. In the DoD, we could not win the nation's wars without them either.

As we look to NASA to field the brainpower and expertise that drives its high-powered, innovative, diverse and multifunctional teams, I remember that as Americans we have an insatiable thirst for technical solutions. I witness it in class every day in my current role as a DAU professor training DoD's acquisition workforce members along their certification pathway.

So, am I concerned about NASA or the acquisition community overcoming their challenges? Not one bit. It's how we are wired as Americans and pioneers, no matter what career we pursue. We just need to make sure we remind ourselves of our potential with some frequency. That and proper funding will keep us unbeatable.

I remember lying on my back in Atlantis 29 years ago going through the countdown check list. As we did in those days, Atlantis launched on time too—a perfect record in my book. Timing is everything; funding is critical—and a little luck helps.

The author can be contacted at **woody.spring@dau.mil**.

Defense AT&L

Writers' Guidelines in Brief

Purpose
Defense AT&L is a bimonthly magazine published by DAU Press, Defense Acquisition University, for senior military personnel, civilians, defense contractors, and defense industry professionals in program management and the acquisition, technology, and logistics workforce.

Submission Procedures
Submit articles by e-mail to datl@dau.mil. Submissions must include each author's name, mailing address, office phone number, e-mail address, and brief biographical statement. Each must also be accompanied by a copyright release.

Receipt of your submission will be acknowledged in 5 working days. You will be notified of our publication decision in 2 to 3 weeks. All decisions are final.

Deadlines
Note: If the magazine fills up before the author deadline, submissions are considered for the following issue.

Issue	**Author Deadline**
January–February	1 October
March–April	1 December
May–June	1 February
July–August	1 April
September–October	1 June
November–December	1 August

Audience
Defense AT&L readers are mainly acquisition professionals serving in career positions covered by the Defense Acquisition Workforce Improvement Act (DAWIA) or industry equivalent.

Style
Defense AT&L prints feature stories focusing on real people and events. The magazine seeks articles that reflect author experiences in and thoughts about acquisition rather than pages of researched information. Articles should discuss the individual's experience with problems and solutions in acquisition, contracting, logistics, or program management, or with emerging trends.

The magazine does not print academic papers; fact sheets; technical papers; white papers; or articles with footnotes, endnotes, or references. Manuscripts meeting any of those criteria are more suitable for DAU's journal, *Defense Acquisition Research Journal (ARJ)*.

Defense AT&L does not reprint from other publications. Please do not submit manuscripts that have appeared elsewhere. *Defense AT&L* does not publish endorsements of products for sale.

Length
Articles should be 1,500–2,500 words.

Format
Send submissions via e-mail as Microsoft Word attachments.

Graphics
Do not embed photographs or charts in the manuscript. Digital files of photos or graphics should be sent as e-mail attachments. **Each figure or chart must be saved as a separate file in the original software format in which it was created.**

TIF or JPEG files must have a resolution of 300 pixels per inch; enhanced resolutions are not acceptable; and images downloaded from the Web are not of adequate quality for reproduction. Detailed tables and charts are not accepted for publication because they will be illegible when reduced to fit at most one-third of a magazine page.

Non-DoD photos and graphics are printed only with written permission from the source. It is the author's responsibility to obtain and submit permission with the article. **Do not include any classified information.**

Author Information
Contact and biographical information will be included with each article selected for publication. Please include the following information with your submission: name, position title, department, institution, address, phone number, and e-mail address. Also, please supply a short biographical statement, not to exceed 25 words. We do not print author bio photographs.

Copyright
All articles require a signed Work of the U.S. Government/Copyright Release form, available at http://www.dau.mil/pubscats/pages/defenseatl.aspx. Fill out, sign, scan, and e-mail it to datl@dau.mil or fax it to 703-805-2917, Attn: Defense AT&L.

Alternatively, you may submit a written release from the major command (normally the public affairs office) indicating the author is releasing the article to *Defense AT&L* for publication without restriction.

The Defense Acquisition University does not accept copyrighted material for publication in *Defense AT&L*. Articles will be considered only if they are unrestricted. This is in keeping with the University's policy that our publications be fully accessible to the public without restriction. All articles are in the public domain and posted to the University's website, www.dau.mil.

http://www.dau.mil/pubscats/pages/defenseatl.aspx

Defense Acquisition University

DEFENSE AT&L

2014 APEX®

AWARDS FOR PUBLICATION EXCELLENCE

Learn. Perform. Succeed.